

general_ledger

Generated by Doxygen 1.8.1.2

Sun Jun 8 2014 21:27:14

Contents

1	General Ledger.	1
2	Data Structure Index	3
2.1	Data Structures	3
3	File Index	5
3.1	File List	5
4	Data Structure Documentation	11
4.1	ds_kvpair Struct Reference	11
4.1.1	Detailed Description	11
4.1.2	Field Documentation	11
4.1.2.1	key	11
4.1.2.2	value	12
4.2	ds_list Struct Reference	12
4.2.1	Detailed Description	12
4.2.2	Field Documentation	12
4.2.2.1	current	12
4.2.2.2	data_destructor	12
4.2.2.3	free_on_delete	13
4.2.2.4	head	13
4.2.2.5	length	13
4.2.2.6	tail	13
4.3	ds_list_element Struct Reference	13
4.3.1	Detailed Description	13
4.3.2	Field Documentation	13
4.3.2.1	data	13
4.3.2.2	next	14
4.3.2.3	previous	14
4.4	ds_map Struct Reference	14
4.4.1	Detailed Description	14
4.4.2	Field Documentation	14

4.4.2.1	hash_size	14
4.4.2.2	lists	14
4.5	ds_map_str Struct Reference	15
4.5.1	Detailed Description	15
4.5.2	Field Documentation	15
4.5.2.1	hash_size	15
4.5.2.2	lists	15
4.6	ds_record Struct Reference	16
4.6.1	Detailed Description	16
4.6.2	Field Documentation	16
4.6.2.1	fields	16
4.7	ds_recordset Struct Reference	17
4.7.1	Detailed Description	17
4.7.2	Field Documentation	17
4.7.2.1	field_lengths	17
4.7.2.2	headers	17
4.7.2.3	num_fields	17
4.7.2.4	records	18
4.7.2.5	types	18
4.8	ds_report Struct Reference	18
4.8.1	Detailed Description	18
4.8.2	Field Documentation	19
4.8.2.1	created_time	19
4.8.2.2	headers	19
4.8.2.3	report_text	19
4.8.2.4	title	19
4.9	ds_str Struct Reference	19
4.9.1	Detailed Description	19
4.9.2	Field Documentation	19
4.9.2.1	capacity	19
4.9.2.2	data	19
4.9.2.3	length	19
4.10	ds_vector Struct Reference	20
4.10.1	Detailed Description	20
4.10.2	Field Documentation	20
4.10.2.1	current	20
4.10.2.2	data	20
4.10.2.3	data_destructor	20
4.10.2.4	free_on_delete	20
4.10.2.5	size	20

4.11	kv_pair_node Struct Reference	21
4.11.1	Detailed Description	21
4.11.2	Field Documentation	21
4.11.2.1	key	21
4.11.2.2	next	21
4.11.2.3	value	21
4.12	params Struct Reference	21
4.12.1	Detailed Description	22
4.12.2	Field Documentation	22
4.12.2.1	database	22
4.12.2.2	hostname	22
4.12.2.3	password	22
4.12.2.4	username	22
5	File Documentation	23
5.1	lib/database/database.h File Reference	23
5.1.1	Detailed Description	24
5.2	lib/database/db_connection.h File Reference	24
5.2.1	Detailed Description	24
5.2.2	Function Documentation	25
5.2.2.1	db_connect	25
5.3	lib/database/db_currenttb.c File Reference	25
5.3.1	Detailed Description	26
5.3.2	Function Documentation	26
5.3.2.1	db_check_total_report	26
5.3.2.2	db_create_check_total_view	26
5.3.2.3	db_create_current_trial_balance_view	26
5.3.2.4	db_current_trial_balance_report	26
5.3.2.5	db_drop_check_total_view	27
5.3.2.6	db_drop_current_trial_balance_view	27
5.4	lib/database/db_currenttb.h File Reference	27
5.4.1	Detailed Description	28
5.4.2	Function Documentation	28
5.4.2.1	db_check_total_report	28
5.4.2.2	db_create_check_total_view	28
5.4.2.3	db_create_current_trial_balance_view	28
5.4.2.4	db_current_trial_balance_report	28
5.4.2.5	db_drop_check_total_view	29
5.4.2.6	db_drop_current_trial_balance_view	29
5.5	lib/database/db_entities.c File Reference	29

5.5.1	Detailed Description	29
5.5.2	Function Documentation	30
5.5.2.1	db_create_entities_table	30
5.5.2.2	db_drop_entities_table	30
5.5.2.3	db_get_entity_name_from_id	30
5.5.2.4	db_list_entities_report	30
5.6	lib/database/db_entities.h File Reference	30
5.6.1	Detailed Description	31
5.6.2	Function Documentation	32
5.6.2.1	db_create_entities_table	32
5.6.2.2	db_drop_entities_table	32
5.6.2.3	db_get_entity_name_from_id	32
5.6.2.4	db_list_entities_report	32
5.7	lib/database/db_internal.h File Reference	33
5.7.1	Detailed Description	33
5.8	lib/database/db_jelines.c File Reference	33
5.8.1	Detailed Description	34
5.8.2	Function Documentation	34
5.8.2.1	db_create_jelines_table	34
5.8.2.2	db_drop_jelines_table	34
5.8.2.3	db_list_jelines_report	35
5.9	lib/database/db_jelines.h File Reference	35
5.9.1	Detailed Description	35
5.9.2	Function Documentation	36
5.9.2.1	db_create_jelines_table	36
5.9.2.2	db_drop_jelines_table	36
5.9.2.3	db_list_jelines_report	36
5.10	lib/database/db_jes.c File Reference	36
5.10.1	Detailed Description	37
5.10.2	Function Documentation	37
5.10.2.1	db_all_jes_report	37
5.10.2.2	db_create_all_jes_view	37
5.10.2.3	db_create_jes_table	37
5.10.2.4	db_drop_all_jes_view	38
5.10.2.5	db_drop_jes_table	38
5.10.2.6	db_list_jes_report	38
5.11	lib/database/db_jes.h File Reference	38
5.11.1	Detailed Description	39
5.11.2	Function Documentation	39
5.11.2.1	db_all_jes_report	39

5.11.2.2	db_create_all_jes_view	39
5.11.2.3	db_create_jes_table	40
5.11.2.4	db_drop_all_jes_view	40
5.11.2.5	db_drop_jes_table	40
5.11.2.6	db_list_jes_report	40
5.12	lib/database/db_jesrcs.c File Reference	40
5.12.1	Detailed Description	41
5.12.2	Function Documentation	41
5.12.2.1	db_create_jesrcs_table	41
5.12.2.2	db_drop_jesrcs_table	41
5.12.2.3	db_list_jesrcs_report	41
5.13	lib/database/db_jesrcs.h File Reference	42
5.13.1	Detailed Description	42
5.13.2	Function Documentation	43
5.13.2.1	db_create_jesrcs_table	43
5.13.2.2	db_drop_jesrcs_table	43
5.13.2.3	db_list_jesrcs_report	43
5.14	lib/database/db_nomaccts.c File Reference	43
5.14.1	Detailed Description	44
5.14.2	Function Documentation	44
5.14.2.1	db_create_nomaccts_table	44
5.14.2.2	db_drop_nomaccts_table	44
5.14.2.3	db_list_nomaccts_report	44
5.15	lib/database/db_nomaccts.h File Reference	44
5.15.1	Detailed Description	45
5.15.2	Function Documentation	45
5.15.2.1	db_create_nomaccts_table	45
5.15.2.2	db_drop_nomaccts_table	46
5.15.2.3	db_list_nomaccts_report	46
5.16	lib/database/db_query.h File Reference	46
5.16.1	Detailed Description	47
5.16.2	Function Documentation	47
5.16.2.1	db_execute_query	47
5.17	lib/database/db_reporting.c File Reference	47
5.17.1	Detailed Description	48
5.17.2	Function Documentation	48
5.17.2.1	db_create_report_from_query	48
5.18	lib/database/db_reporting.h File Reference	48
5.18.1	Detailed Description	48
5.18.2	Function Documentation	49

5.18.2.1 db_create_recordset_from_query	49
5.18.2.2 db_create_report_from_query	49
5.19 lib/database/db_sampledata.c File Reference	49
5.19.1 Detailed Description	50
5.20 lib/database/db_sampledata.h File Reference	50
5.20.1 Detailed Description	51
5.21 lib/database/db_sql.h File Reference	51
5.21.1 Detailed Description	52
5.21.2 Function Documentation	53
5.21.2.1 db_all_jes_number_report_sql	53
5.21.2.2 db_all_jes_report_sql	53
5.21.2.3 db_check_total_entity_report_sql	53
5.21.2.4 db_check_total_report_sql	53
5.21.2.5 db_create_all_jes_view_sql	53
5.21.2.6 db_create_check_total_view_sql	53
5.21.2.7 db_create_current_trial_balance_view_sql	54
5.21.2.8 db_create_entities_table_sql	54
5.21.2.9 db_create_jelines_table_sql	54
5.21.2.10 db_create_jes_table_sql	54
5.21.2.11 db_create_jesrcs_table_sql	54
5.21.2.12 db_create_nomaccts_table_sql	54
5.21.2.13 db_create_standingdata_table_sql	54
5.21.2.14 db_create_users_table_sql	55
5.21.2.15 db_current_trial_balance_entity_report_sql	55
5.21.2.16 db_current_trial_balance_report_sql	55
5.21.2.17 db_drop_all_jes_view_sql	55
5.21.2.18 db_drop_check_total_view_sql	55
5.21.2.19 db_drop_current_trial_balance_view_sql	55
5.21.2.20 db_drop_entities_table_sql	55
5.21.2.21 db_drop_jelines_table_sql	56
5.21.2.22 db_drop_jes_table_sql	56
5.21.2.23 db_drop_jesrcs_table_sql	56
5.21.2.24 db_drop_nomaccts_table_sql	56
5.21.2.25 db_drop_standingdata_table_sql	56
5.21.2.26 db_drop_users_table_sql	56
5.21.2.27 db_get_entity_name_from_id_sql	56
5.21.2.28 db_list_entities_report_sql	57
5.21.2.29 db_list_jelines_report_sql	57
5.21.2.30 db_list_jes_report_sql	57
5.21.2.31 db_list_jesrcs_report_sql	57

5.21.2.32 db_list_nomaccts_report_sql	57
5.21.2.33 db_list_users_report_sql	57
5.21.2.34 db_show_standingdata_report_sql	57
5.22 lib/database/db_standingdata.c File Reference	58
5.22.1 Detailed Description	58
5.22.2 Function Documentation	58
5.22.2.1 db_create_standingdata_table	58
5.22.2.2 db_drop_standingdata_table	59
5.22.2.3 db_show_standingdata_report	59
5.23 lib/database/db_standingdata.h File Reference	59
5.23.1 Detailed Description	60
5.23.2 Function Documentation	60
5.23.2.1 db_create_standingdata_table	60
5.23.2.2 db_drop_standingdata_table	60
5.23.2.3 db_show_standingdata_report	60
5.24 lib/database/db_structure.c File Reference	60
5.24.1 Detailed Description	61
5.24.2 Function Documentation	61
5.24.2.1 db_create_database_structure	61
5.24.2.2 db_delete_database_structure	61
5.25 lib/database/db_structure.h File Reference	62
5.25.1 Detailed Description	62
5.25.2 Function Documentation	62
5.25.2.1 db_create_database_structure	62
5.25.2.2 db_delete_database_structure	63
5.26 lib/database/db_users.c File Reference	63
5.26.1 Detailed Description	63
5.26.2 Function Documentation	64
5.26.2.1 db_create_users_table	64
5.26.2.2 db_drop_users_table	64
5.26.2.3 db_list_users_report	64
5.27 lib/database/db_users.h File Reference	64
5.27.1 Detailed Description	65
5.27.2 Function Documentation	65
5.27.2.1 db_create_users_table	65
5.27.2.2 db_drop_users_table	65
5.27.2.3 db_list_users_report	65
5.28 lib/database/dummy/db_dummy_create_entities_table_sql.c File Reference	65
5.28.1 Detailed Description	66
5.28.2 Function Documentation	66

5.28.2.1	db_create_entities_table_sql	66
5.29	lib/database/dummy/db_dummy_create_users_table_sql.c File Reference	66
5.29.1	Detailed Description	66
5.29.2	Function Documentation	67
5.29.2.1	db_create_users_table_sql	67
5.30	lib/database/dummy/db_dummy_drop_entities_table_sql.c File Reference	67
5.30.1	Detailed Description	67
5.30.2	Function Documentation	67
5.30.2.1	db_drop_entities_table_sql	67
5.31	lib/database/dummy/db_dummy_drop_users_table_sql.c File Reference	67
5.31.1	Detailed Description	68
5.31.2	Function Documentation	68
5.31.2.1	db_drop_users_table_sql	68
5.32	lib/database/dummy/db_dummy_general.c File Reference	68
5.32.1	Detailed Description	69
5.32.2	Macro Definition Documentation	69
5.32.2.1	_XOPEN_SOURCE	69
5.32.3	Function Documentation	69
5.32.3.1	db_connect	69
5.32.3.2	db_create_recordset_from_query	69
5.32.3.3	db_execute_query	70
5.33	lib/database/dummy/db_dummy_list_entities_report_sql.c File Reference	70
5.33.1	Detailed Description	70
5.33.2	Function Documentation	70
5.33.2.1	db_list_entities_report_sql	70
5.34	lib/database/dummy/db_dummy_list_users_report_sql.c File Reference	71
5.34.1	Detailed Description	71
5.34.2	Function Documentation	71
5.34.2.1	db_list_users_report_sql	71
5.35	lib/database/mysql/db_mysql_all_jes_number_report_sql.c File Reference	71
5.35.1	Detailed Description	71
5.35.2	Function Documentation	72
5.35.2.1	db_all_jes_number_report_sql	72
5.36	lib/database/mysql/db_mysql_all_jes_report_sql.c File Reference	72
5.36.1	Detailed Description	72
5.36.2	Function Documentation	72
5.36.2.1	db_all_jes_report_sql	72
5.37	lib/database/mysql/db_mysql_check_total_entity_report_sql.c File Reference	72
5.37.1	Detailed Description	73
5.37.2	Function Documentation	73

5.37.2.1 db_check_total_entity_report_sql	73
5.38 lib/database/mysql/db_mysql_check_total_report_sql.c File Reference	73
5.38.1 Detailed Description	73
5.38.2 Function Documentation	73
5.38.2.1 db_check_total_report_sql	73
5.39 lib/database/mysql/db_mysql_create_all_jes_view_sql.c File Reference	74
5.39.1 Detailed Description	74
5.39.2 Function Documentation	74
5.39.2.1 db_create_all_jes_view_sql	74
5.40 lib/database/mysql/db_mysql_create_check_total_view_sql.c File Reference	74
5.40.1 Detailed Description	74
5.40.2 Function Documentation	75
5.40.2.1 db_create_check_total_view_sql	75
5.41 lib/database/mysql/db_mysql_create_current_trial_balance_view_sql.c File Reference	75
5.41.1 Detailed Description	75
5.41.2 Function Documentation	75
5.41.2.1 db_create_current_trial_balance_view_sql	75
5.42 lib/database/mysql/db_mysql_create_entities_table_sql.c File Reference	75
5.42.1 Detailed Description	76
5.42.2 Function Documentation	76
5.42.2.1 db_create_entities_table_sql	76
5.43 lib/database/mysql/db_mysql_create_jelines_table_sql.c File Reference	76
5.43.1 Detailed Description	76
5.43.2 Function Documentation	76
5.43.2.1 db_create_jelines_table_sql	76
5.44 lib/database/mysql/db_mysql_create_jes_table_sql.c File Reference	77
5.44.1 Detailed Description	77
5.44.2 Function Documentation	77
5.44.2.1 db_create_jes_table_sql	77
5.45 lib/database/mysql/db_mysql_create_jesrcs_table_sql.c File Reference	77
5.45.1 Detailed Description	77
5.45.2 Function Documentation	78
5.45.2.1 db_create_jesrcs_table_sql	78
5.46 lib/database/mysql/db_mysql_create_nomaccts_table_sql.c File Reference	78
5.46.1 Detailed Description	78
5.46.2 Function Documentation	78
5.46.2.1 db_create_nomaccts_table_sql	78
5.47 lib/database/mysql/db_mysql_create_standingdata_table_sql.c File Reference	78
5.47.1 Detailed Description	79
5.47.2 Function Documentation	79

5.47.2.1 db_create_standingdata_table_sql	79
5.48 lib/database/mysql/db_mysql_create_users_table_sql.c File Reference	79
5.48.1 Detailed Description	79
5.48.2 Function Documentation	79
5.48.2.1 db_create_users_table_sql	79
5.49 lib/database/mysql/db_mysql_current_trial_balance_entity_report_sql.c File Reference	80
5.49.1 Detailed Description	80
5.49.2 Function Documentation	80
5.49.2.1 db_current_trial_balance_entity_report_sql	80
5.50 lib/database/mysql/db_mysql_current_trial_balance_report_sql.c File Reference	80
5.50.1 Detailed Description	80
5.50.2 Function Documentation	81
5.50.2.1 db_current_trial_balance_report_sql	81
5.51 lib/database/mysql/db_mysql_drop_all_jes_view_sql.c File Reference	81
5.51.1 Detailed Description	81
5.51.2 Function Documentation	81
5.51.2.1 db_drop_all_jes_view_sql	81
5.52 lib/database/mysql/db_mysql_drop_check_total_view_sql.c File Reference	81
5.52.1 Detailed Description	82
5.52.2 Function Documentation	82
5.52.2.1 db_drop_check_total_view_sql	82
5.53 lib/database/mysql/db_mysql_drop_current_trial_balance_view_sql.c File Reference	82
5.53.1 Detailed Description	82
5.53.2 Function Documentation	82
5.53.2.1 db_drop_current_trial_balance_view_sql	82
5.54 lib/database/mysql/db_mysql_drop_entities_table_sql.c File Reference	83
5.54.1 Detailed Description	83
5.54.2 Function Documentation	83
5.54.2.1 db_drop_entities_table_sql	83
5.55 lib/database/mysql/db_mysql_drop_jelines_table_sql.c File Reference	83
5.55.1 Detailed Description	83
5.55.2 Function Documentation	84
5.55.2.1 db_drop_jelines_table_sql	84
5.56 lib/database/mysql/db_mysql_drop_jes_table_sql.c File Reference	84
5.56.1 Detailed Description	84
5.56.2 Function Documentation	84
5.56.2.1 db_drop_jes_table_sql	84
5.57 lib/database/mysql/db_mysql_drop_jesrcs_table_sql.c File Reference	84
5.57.1 Detailed Description	85
5.57.2 Function Documentation	85

5.57.2.1 db_drop_jesrcs_table_sql	85
5.58 lib/database/mysql/db_mysql_drop_nomaccts_table_sql.c File Reference	85
5.58.1 Detailed Description	85
5.58.2 Function Documentation	85
5.58.2.1 db_drop_nomaccts_table_sql	85
5.59 lib/database/mysql/db_mysql_drop_standingdata_table_sql.c File Reference	86
5.59.1 Detailed Description	86
5.59.2 Function Documentation	86
5.59.2.1 db_drop_standingdata_table_sql	86
5.60 lib/database/mysql/db_mysql_drop_users_table_sql.c File Reference	86
5.60.1 Detailed Description	86
5.60.2 Function Documentation	87
5.60.2.1 db_drop_users_table_sql	87
5.61 lib/database/mysql/db_mysql_entity_name_from_id_sql.c File Reference	87
5.61.1 Detailed Description	87
5.61.2 Function Documentation	87
5.61.2.1 db_get_entity_name_from_id_sql	87
5.62 lib/database/mysql/db_mysql_general.c File Reference	87
5.62.1 Detailed Description	88
5.62.2 Function Documentation	88
5.62.2.1 db_connect	88
5.62.2.2 db_create_recordset_from_query	89
5.62.2.3 db_execute_query	89
5.62.3 Variable Documentation	89
5.62.3.1 conn_mss	89
5.62.3.2 main_mss	89
5.63 lib/database/mysql/db_mysql_list_entities_report_sql.c File Reference	89
5.63.1 Detailed Description	89
5.63.2 Function Documentation	90
5.63.2.1 db_list_entities_report_sql	90
5.64 lib/database/mysql/db_mysql_list_jelines_report_sql.c File Reference	90
5.64.1 Detailed Description	90
5.64.2 Function Documentation	90
5.64.2.1 db_list_jelines_report_sql	90
5.65 lib/database/mysql/db_mysql_list_jes_report_sql.c File Reference	90
5.65.1 Detailed Description	91
5.65.2 Function Documentation	91
5.65.2.1 db_list_jes_report_sql	91
5.66 lib/database/mysql/db_mysql_list_jesrcs_report_sql.c File Reference	91
5.66.1 Detailed Description	91

5.66.2	Function Documentation	92
5.66.2.1	db_list_jesrcs_report_sql	92
5.67	lib/database/mysql/db_mysql_list_nomaccts_report_sql.c File Reference	92
5.67.1	Detailed Description	92
5.67.2	Function Documentation	92
5.67.2.1	db_list_nomaccts_report_sql	92
5.68	lib/database/mysql/db_mysql_list_users_report_sql.c File Reference	92
5.68.1	Detailed Description	93
5.68.2	Function Documentation	93
5.68.2.1	db_list_users_report_sql	93
5.69	lib/database/mysql/db_mysql_show_standingdata_report_sql.c File Reference	93
5.69.1	Detailed Description	93
5.69.2	Function Documentation	93
5.69.2.1	db_show_standingdata_report_sql	93
5.70	lib/datastruct/data_structures.h File Reference	94
5.70.1	Detailed Description	94
5.71	lib/datastruct/ds_fieldtypes.h File Reference	94
5.71.1	Detailed Description	95
5.71.2	Enumeration Type Documentation	95
5.71.2.1	ds_field_types	95
5.72	lib/datastruct/ds_kvpair.h File Reference	95
5.72.1	Detailed Description	96
5.72.2	Typedef Documentation	97
5.72.2.1	ds_kvpair	97
5.72.3	Function Documentation	97
5.72.3.1	ds_kvpair_create	97
5.72.3.2	ds_kvpair_destroy	97
5.72.3.3	ds_kvpair_destructor	97
5.72.3.4	ds_kvpair_get_key	98
5.72.3.5	ds_kvpair_get_value	98
5.73	lib/datastruct/ds_list.c File Reference	98
5.73.1	Detailed Description	99
5.73.2	Function Documentation	99
5.73.2.1	ds_list_append	99
5.73.2.2	ds_list_create	100
5.73.2.3	ds_list_destroy	100
5.73.2.4	ds_list_destructor	100
5.73.2.5	ds_list_element	100
5.73.2.6	ds_list_get_next_data	101
5.73.2.7	ds_list_get_prev_data	101

5.73.2.8	ds_list_is_empty	101
5.73.2.9	ds_list_length	101
5.73.2.10	ds_list_remove_all	102
5.73.2.11	ds_list_remove_tail	102
5.73.2.12	ds_list_seek_end	102
5.73.2.13	ds_list_seek_start	102
5.74	lib/datastruct/ds_list.h File Reference	102
5.74.1	Detailed Description	103
5.74.2	Typedef Documentation	104
5.74.2.1	ds_list	104
5.74.3	Function Documentation	104
5.74.3.1	ds_list_append	104
5.74.3.2	ds_list_create	104
5.74.3.3	ds_list_destroy	104
5.74.3.4	ds_list_destructor	104
5.74.3.5	ds_list_element	105
5.74.3.6	ds_list_get_next_data	105
5.74.3.7	ds_list_get_prev_data	105
5.74.3.8	ds_list_is_empty	105
5.74.3.9	ds_list_length	106
5.74.3.10	ds_list_remove_all	106
5.74.3.11	ds_list_remove_tail	106
5.74.3.12	ds_list_seek_end	106
5.74.3.13	ds_list_seek_start	106
5.75	lib/datastruct/ds_map.c File Reference	106
5.75.1	Detailed Description	107
5.75.2	Function Documentation	108
5.75.2.1	ds_map_destroy	108
5.75.2.2	ds_map_get_value	108
5.75.2.3	ds_map_init	108
5.75.2.4	ds_map_insert	108
5.75.2.5	ds_map_print_all	109
5.76	lib/datastruct/ds_map.h File Reference	109
5.76.1	Detailed Description	110
5.76.2	Typedef Documentation	110
5.76.2.1	ds_map	110
5.76.3	Function Documentation	110
5.76.3.1	ds_map_destroy	110
5.76.3.2	ds_map_get_value	110
5.76.3.3	ds_map_init	110

5.76.3.4	ds_map_insert	111
5.76.3.5	ds_map_print_all	111
5.77	lib/datastruct/ds_map_str.c File Reference	111
5.77.1	Detailed Description	112
5.77.2	Function Documentation	112
5.77.2.1	ds_map_str_destroy	112
5.77.2.2	ds_map_str_get_value	112
5.77.2.3	ds_map_str_init	113
5.77.2.4	ds_map_str_insert	113
5.78	lib/datastruct/ds_map_str.h File Reference	113
5.78.1	Detailed Description	114
5.78.2	Typedef Documentation	114
5.78.2.1	ds_map_str	114
5.78.3	Function Documentation	114
5.78.3.1	ds_map_str_destroy	114
5.78.3.2	ds_map_str_get_value	115
5.78.3.3	ds_map_str_init	115
5.78.3.4	ds_map_str_insert	115
5.79	lib/datastruct/ds_record.c File Reference	115
5.79.1	Detailed Description	116
5.79.2	Function Documentation	117
5.79.2.1	ds_record_clear	117
5.79.2.2	ds_record_create	117
5.79.2.3	ds_record_destroy	117
5.79.2.4	ds_record_destructor	117
5.79.2.5	ds_record_get_field	117
5.79.2.6	ds_record_get_next_data	118
5.79.2.7	ds_record_make_delim_string	118
5.79.2.8	ds_record_make_values_string	118
5.79.2.9	ds_record_seek_start	118
5.79.2.10	ds_record_set_field	119
5.79.2.11	ds_record_size	119
5.79.2.12	ds_record_tokenize	119
5.80	lib/datastruct/ds_record.h File Reference	119
5.80.1	Detailed Description	121
5.80.2	Typedef Documentation	121
5.80.2.1	ds_record	121
5.80.3	Function Documentation	121
5.80.3.1	ds_record_clear	121
5.80.3.2	ds_record_create	121

5.80.3.3	ds_record_destroy	121
5.80.3.4	ds_record_destructor	122
5.80.3.5	ds_record_get_field	122
5.80.3.6	ds_record_get_next_data	122
5.80.3.7	ds_record_make_delim_string	122
5.80.3.8	ds_record_make_values_string	123
5.80.3.9	ds_record_seek_start	123
5.80.3.10	ds_record_set_field	123
5.80.3.11	ds_record_size	123
5.80.3.12	ds_record_tokenize	123
5.81	lib/datastruct/ds_recordset.c File Reference	124
5.81.1	Detailed Description	125
5.81.2	Function Documentation	125
5.81.2.1	ds_recordset_add_record	125
5.81.2.2	ds_recordset_create	125
5.81.2.3	ds_recordset_destroy	125
5.81.2.4	ds_recordset_get_next_insert_query	126
5.81.2.5	ds_recordset_get_text_report	126
5.81.2.6	ds_recordset_next_record	126
5.81.2.7	ds_recordset_num_fields	126
5.81.2.8	ds_recordset_num_records	127
5.81.2.9	ds_recordset_seek_start	127
5.81.2.10	ds_recordset_set_headers	127
5.81.2.11	ds_recordset_set_type	127
5.82	lib/datastruct/ds_recordset.h File Reference	127
5.82.1	Detailed Description	129
5.82.2	Typedef Documentation	129
5.82.2.1	ds_recordset	129
5.82.3	Function Documentation	129
5.82.3.1	ds_recordset_add_record	129
5.82.3.2	ds_recordset_create	129
5.82.3.3	ds_recordset_destroy	130
5.82.3.4	ds_recordset_get_next_insert_query	130
5.82.3.5	ds_recordset_get_text_report	130
5.82.3.6	ds_recordset_next_record	130
5.82.3.7	ds_recordset_num_fields	131
5.82.3.8	ds_recordset_num_records	131
5.82.3.9	ds_recordset_seek_start	131
5.82.3.10	ds_recordset_set_headers	131
5.82.3.11	ds_recordset_set_type	131

5.83	lib/datastruct/ds_report.c File Reference	132
5.83.1	Detailed Description	132
5.83.2	Macro Definition Documentation	133
5.83.2.1	_XOPEN_SOURCE	133
5.83.3	Function Documentation	133
5.83.3.1	ds_report_add_header	133
5.83.3.2	ds_report_create	133
5.83.3.3	ds_report_destroy	133
5.83.3.4	ds_report_print_text_report	133
5.83.3.5	ds_report_set_report_text	134
5.83.3.6	ds_report_set_title	134
5.84	lib/datastruct/ds_report.h File Reference	134
5.84.1	Detailed Description	135
5.84.2	Typedef Documentation	135
5.84.2.1	ds_report	135
5.84.3	Function Documentation	135
5.84.3.1	ds_report_add_header	135
5.84.3.2	ds_report_create	136
5.84.3.3	ds_report_destroy	136
5.84.3.4	ds_report_print_text_report	136
5.84.3.5	ds_report_set_report_text	136
5.84.3.6	ds_report_set_title	136
5.85	lib/datastruct/ds_str.c File Reference	137
5.85.1	Detailed Description	139
5.85.2	Function Documentation	139
5.85.2.1	ds_str_assign	139
5.85.2.2	ds_str_assign_cstr	139
5.85.2.3	ds_str_char_at_index	139
5.85.2.4	ds_str_clear	140
5.85.2.5	ds_str_compare	140
5.85.2.6	ds_str_compare_cstr	140
5.85.2.7	ds_str_concat	140
5.85.2.8	ds_str_concat_cstr	140
5.85.2.9	ds_str_create	141
5.85.2.10	ds_str_create_direct	141
5.85.2.11	ds_str_create_sprintf	141
5.85.2.12	ds_str_cstr	142
5.85.2.13	ds_str_decorate	142
5.85.2.14	ds_str_destroy	142
5.85.2.15	ds_str_destructor	142

5.85.2.16 ds_str_doubleval	142
5.85.2.17 ds_str_dup	143
5.85.2.18 ds_str_getline	143
5.85.2.19 ds_str_hash	143
5.85.2.20 ds_str_intval	143
5.85.2.21 ds_str_is_alnum	144
5.85.2.22 ds_str_is_empty	144
5.85.2.23 ds_str_length	144
5.85.2.24 ds_str_size_to_fit	144
5.85.2.25 ds_str_split	145
5.85.2.26 ds_str_strchr	145
5.85.2.27 ds_str_substr_left	145
5.85.2.28 ds_str_substr_right	145
5.85.2.29 ds_str_trim	146
5.85.2.30 ds_str_trim_leading	146
5.85.2.31 ds_str_trim_trailing	146
5.85.2.32 ds_str_trunc	146
5.86 lib/datastruct/ds_str.h File Reference	146
5.86.1 Detailed Description	149
5.86.2 Typedef Documentation	149
5.86.2.1 ds_str	149
5.86.3 Function Documentation	149
5.86.3.1 ds_str_assign	149
5.86.3.2 ds_str_assign_cstr	149
5.86.3.3 ds_str_char_at_index	149
5.86.3.4 ds_str_clear	150
5.86.3.5 ds_str_compare	150
5.86.3.6 ds_str_compare_cstr	150
5.86.3.7 ds_str_concat	150
5.86.3.8 ds_str_concat_cstr	151
5.86.3.9 ds_str_create	151
5.86.3.10 ds_str_create_direct	151
5.86.3.11 ds_str_create_sprintf	151
5.86.3.12 ds_str_cstr	152
5.86.3.13 ds_str_decorate	152
5.86.3.14 ds_str_destroy	152
5.86.3.15 ds_str_destructor	152
5.86.3.16 ds_str_doubleval	152
5.86.3.17 ds_str_dup	153
5.86.3.18 ds_str_getline	153

5.86.3.19 ds_str_hash	153
5.86.3.20 ds_str_intval	153
5.86.3.21 ds_str_is_alnum	154
5.86.3.22 ds_str_is_empty	154
5.86.3.23 ds_str_length	154
5.86.3.24 ds_str_size_to_fit	154
5.86.3.25 ds_str_split	155
5.86.3.26 ds_str_strchr	155
5.86.3.27 ds_str_substr_left	155
5.86.3.28 ds_str_substr_right	155
5.86.3.29 ds_str_trim	156
5.86.3.30 ds_str_trim_leading	156
5.86.3.31 ds_str_trim_trailing	156
5.86.3.32 ds_str_trunc	156
5.87 lib/datastruct/ds_vector.c File Reference	156
5.87.1 Detailed Description	157
5.87.2 Function Documentation	158
5.87.2.1 ds_vector_clear	158
5.87.2.2 ds_vector_create	158
5.87.2.3 ds_vector_destroy	158
5.87.2.4 ds_vector_destructor	158
5.87.2.5 ds_vector_element	159
5.87.2.6 ds_vector_get_next_data	159
5.87.2.7 ds_vector_seek_start	159
5.87.2.8 ds_vector_set	159
5.87.2.9 ds_vector_size	159
5.88 lib/datastruct/ds_vector.h File Reference	160
5.88.1 Detailed Description	161
5.88.2 Typedef Documentation	161
5.88.2.1 ds_vector	161
5.88.3 Function Documentation	161
5.88.3.1 ds_vector_clear	161
5.88.3.2 ds_vector_create	161
5.88.3.3 ds_vector_destroy	162
5.88.3.4 ds_vector_destructor	162
5.88.3.5 ds_vector_element	162
5.88.3.6 ds_vector_get_next_data	162
5.88.3.7 ds_vector_seek_start	163
5.88.3.8 ds_vector_set	163
5.88.3.9 ds_vector_size	163

5.89 lib/file_ops/config_file_read.c File Reference	163
5.89.1 Detailed Description	164
5.89.2 Macro Definition Documentation	165
5.89.2.1 CONFIG_MAP_SIZE	165
5.89.2.2 MAX_BUFFER_SIZE	165
5.89.3 Function Documentation	165
5.89.3.1 config_file_read	165
5.89.3.2 config_free	165
5.89.3.3 config_init	165
5.89.3.4 config_value_get	165
5.89.3.5 config_value_get_cstr	166
5.89.3.6 config_value_set	166
5.90 lib/file_ops/config_file_read.h File Reference	166
5.90.1 Detailed Description	167
5.90.2 Macro Definition Documentation	167
5.90.2.1 CONFIG_FILE_MALFORMED_FILE	167
5.90.2.2 CONFIG_FILE_NO_FILE	168
5.90.2.3 CONFIG_FILE_OK	168
5.90.3 Function Documentation	168
5.90.3.1 config_file_read	168
5.90.3.2 config_free	168
5.90.3.3 config_init	168
5.90.3.4 config_value_get	168
5.90.3.5 config_value_get_cstr	169
5.90.3.6 config_value_set	169
5.91 lib/file_ops/delim_file_read.c File Reference	169
5.91.1 Detailed Description	170
5.91.2 Macro Definition Documentation	170
5.91.2.1 MAX_LINE_SIZE	170
5.91.3 Function Documentation	170
5.91.3.1 delim_file_read	170
5.92 lib/file_ops/delim_file_read.h File Reference	171
5.92.1 Detailed Description	171
5.92.2 Function Documentation	172
5.92.2.1 delim_file_read	172
5.93 lib/file_ops/file_ops.h File Reference	172
5.93.1 Detailed Description	173
5.94 lib/gl_general/gl_config.c File Reference	173
5.94.1 Detailed Description	173
5.94.2 Function Documentation	174

5.94.2.1	get_configuration	174
5.94.2.2	params_free	174
5.94.2.3	params_init	174
5.95	lib/gl_general/gl_config.h File Reference	175
5.95.1	Detailed Description	175
5.95.2	Function Documentation	176
5.95.2.1	get_configuration	176
5.95.2.2	params_free	176
5.95.2.3	params_init	176
5.96	lib/gl_general/gl_errors.c File Reference	177
5.96.1	Detailed Description	177
5.96.2	Function Documentation	177
5.96.2.1	gl_error_quit	177
5.97	lib/gl_general/gl_errors.h File Reference	178
5.97.1	Detailed Description	178
5.97.2	Function Documentation	178
5.97.2.1	gl_error_quit	178
5.98	lib/gl_general/gl_general.h File Reference	178
5.98.1	Detailed Description	179
5.99	lib/gl_general/gl_logging.c File Reference	179
5.99.1	Detailed Description	180
5.99.2	Function Documentation	180
5.99.2.1	gl_log_msg	180
5.99.2.2	gl_set_logging	180
5.100	lib/gl_general/gl_logging.h File Reference	181
5.100.1	Detailed Description	181
5.100.2	Function Documentation	182
5.100.2.1	gl_log_msg	182
5.100.2.2	gl_set_logging	182
5.101	lib/gl_general/gl_login.c File Reference	182
5.101.1	Detailed Description	183
5.101.2	Function Documentation	183
5.101.2.1	login	183
5.102	lib/gl_general/gl_login.h File Reference	183
5.102.1	Detailed Description	184
5.102.2	Function Documentation	184
5.102.2.1	login	184
5.103	progs/gl_db/gl_db_config.c File Reference	184
5.103.1	Detailed Description	185
5.103.2	Macro Definition Documentation	185

5.103.2.1 <code>_XOPEN_SOURCE</code>	185
5.103.3 Function Documentation	186
5.103.3.1 <code>get_cmdline_options</code>	186
5.104 <code>progs/gl_db/gl_db_config.h</code> File Reference	186
5.104.1 Detailed Description	187
5.104.2 Function Documentation	187
5.104.2.1 <code>get_cmdline_options</code>	187
5.105 <code>progs/gl_db/gl_db_main.c</code> File Reference	188
5.105.1 Detailed Description	188
5.105.2 Function Documentation	189
5.105.2.1 <code>main</code>	189
5.105.2.2 <code>print_help_message</code>	189
5.105.2.3 <code>print_usage_message</code>	189
5.105.2.4 <code>print_version_message</code>	189
5.106 <code>progs/gl_reports/gl_reports_config.c</code> File Reference	189
5.106.1 Detailed Description	190
5.106.2 Macro Definition Documentation	190
5.106.2.1 <code>_XOPEN_SOURCE</code>	190
5.106.3 Function Documentation	191
5.106.3.1 <code>get_cmdline_options</code>	191
5.107 <code>progs/gl_reports/gl_reports_config.h</code> File Reference	191
5.107.1 Detailed Description	192
5.107.2 Function Documentation	192
5.107.2.1 <code>get_cmdline_options</code>	192

Chapter 1

General Ledger.

General Ledger will be a fully-featured, multi-user, open-source general ledger system. The project is in the early stages of development.

Chapter 2

Data Structure Index

2.1 Data Structures

Here are the data structures with brief descriptions:

ds_kvpair	11
ds_list	12
ds_list_element	13
ds_map	14
ds_map_str	15
ds_record	16
ds_recordset	17
ds_report	18
ds_str	19
ds_vector	20
kv_pair_node	21
params	21

Chapter 3

File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

lib/database/ database.h	
User interface to database functionality	23
lib/database/ db_connection.h	
Interface to database connection functionality	24
lib/database/ db_currenttb.c	
Implementation of current TB functionality	25
lib/database/ db_currenttb.h	
Interface to current trial balance functionality	27
lib/database/ db_entities.c	
Implementation of entities functionality	29
lib/database/ db_entities.h	
Interface to entities functionality	30
lib/database/ db_internal.h	
Internal library interface to database functionality	33
lib/database/ db_jelines.c	
Implementation of journal entries functionality	33
lib/database/ db_jelines.h	
Interface to journal entry lines functionality	35
lib/database/ db_jes.c	
Implementation of journal entries functionality	36
lib/database/ db_jes.h	
Interface to journal entries functionality	38
lib/database/ db_jesrcs.c	
Implementation of journal entry sources functionality	40
lib/database/ db_jesrcs.h	
Interface to journal entry sources functionality	42
lib/database/ db_nomaccts.c	
Implementation of nominal accounts functionality	43
lib/database/ db_nomaccts.h	
Interface to nominal accounts functionality	44
lib/database/ db_query.h	
Interface to database query functionality	46
lib/database/ db_reporting.c	
Implementation of database reporting functionality	47
lib/database/ db_reporting.h	
Interface to database reporting functionality	48
lib/database/ db_sampledata.c	
Implementation of database sample data functionality	49

lib/database/db_sampledata.h	
Interface to database sample data functionality	50
lib/database/db_sql.h	
Interface to database specific SQL strings	51
lib/database/db_standingdata.c	
Implementation of standing data functionality	58
lib/database/db_standingdata.h	
Interface to journal entries functionality	59
lib/database/db_structure.c	
Implementation of database structure functionality	60
lib/database/db_structure.h	
Interface to database structure functionality	62
lib/database/db_users.c	
Implementation of users functionality	63
lib/database/db_users.h	
Interface to users functionality	64
lib/database/dummy/db_dummy_create_entities_table_sql.c	
Returns dummy SQL query to create entities table	65
lib/database/dummy/db_dummy_create_users_table_sql.c	
Returns dummy SQL query to create users table	66
lib/database/dummy/db_dummy_drop_entities_table_sql.c	
Returns dummy SQL query to drop entities table	67
lib/database/dummy/db_dummy_drop_users_table_sql.c	
Returns dummy SQL query to drop users table	67
lib/database/dummy/db_dummy_general.c	
Implementation of dummy database functionality	68
lib/database/dummy/db_dummy_list_entities_report_sql.c	
Returns dummy SQL query to create list entities report	70
lib/database/dummy/db_dummy_list_users_report_sql.c	
Returns dummy SQL query to create list users report	71
lib/database/mysql/db_mysql_all_jes_number_report_sql.c	
Returns MYSQL SQL query to create JE by number report	71
lib/database/mysql/db_mysql_all_jes_report_sql.c	
Returns MYSQL SQL query to create all jes report	72
lib/database/mysql/db_mysql_check_total_entity_report_sql.c	
Returns MYSQL SQL query to create check total report by entity	72
lib/database/mysql/db_mysql_check_total_report_sql.c	
Returns MYSQL SQL query to create check total report	73
lib/database/mysql/db_mysql_create_all_jes_view_sql.c	
Returns MYSQL SQL query to create all_jes view	74
lib/database/mysql/db_mysql_create_check_total_view_sql.c	
Returns MYSQL SQL query to create check total view	74
lib/database/mysql/db_mysql_create_current_trial_balance_view_sql.c	
Returns MYSQL SQL query to create trial balance view	75
lib/database/mysql/db_mysql_create_entities_table_sql.c	
Returns MYSQL SQL query to create entities table	75
lib/database/mysql/db_mysql_create_jelines_table_sql.c	
Returns MYSQL SQL query to create journal entry lines table	76
lib/database/mysql/db_mysql_create_jes_table_sql.c	
Returns MYSQL SQL query to create journal entries table	77
lib/database/mysql/db_mysql_create_jesrcs_table_sql.c	
Returns MYSQL SQL query to create JE sources table	77
lib/database/mysql/db_mysql_create_nomaccts_table_sql.c	
Returns MYSQL SQL query to create nominal accounts table	78
lib/database/mysql/db_mysql_create_standingdata_table_sql.c	
Returns MYSQL SQL query to create standing data table	78
lib/database/mysql/db_mysql_create_users_table_sql.c	
Returns MYSQL SQL query to create users table	79

lib/database/mysql/db_mysql_current_trial_balance_entity_report_sql.c	
Returns MYSQL SQL query to create current TB by entity report	80
lib/database/mysql/db_mysql_current_trial_balance_report_sql.c	
Returns MYSQL SQL query to create current TB report	80
lib/database/mysql/db_mysql_drop_all_jes_view_sql.c	
Returns MYSQL SQL query to drop all JES view	81
lib/database/mysql/db_mysql_drop_check_total_view_sql.c	
Returns MYSQL SQL query to drop check total view	81
lib/database/mysql/db_mysql_drop_current_trial_balance_view_sql.c	
Returns MYSQL SQL query to drop trial balance view	82
lib/database/mysql/db_mysql_drop_entities_table_sql.c	
Returns MYSQL SQL query to drop entities table	83
lib/database/mysql/db_mysql_drop_jelines_table_sql.c	
Returns MYSQL SQL query to drop journal entry lines table	83
lib/database/mysql/db_mysql_drop_jes_table_sql.c	
Returns MYSQL SQL query to drop entities table	84
lib/database/mysql/db_mysql_drop_jesrcs_table_sql.c	
Returns MYSQL SQL query to drop JE sources table	84
lib/database/mysql/db_mysql_drop_nomaccts_table_sql.c	
Returns MYSQL SQL query to drop nominal accounts table	85
lib/database/mysql/db_mysql_drop_standingdata_table_sql.c	
Returns MYSQL SQL query to drop standing data table	86
lib/database/mysql/db_mysql_drop_users_table_sql.c	
Returns MYSQL SQL query to drop users table	86
lib/database/mysql/db_mysql_entity_name_from_id_sql.c	
Returns MYSQL SQL query to fetch an entity name from its ID	87
lib/database/mysql/db_mysql_general.c	
Implementation of MYSQL database functionality	87
lib/database/mysql/db_mysql_list_entities_report_sql.c	
Returns MYSQL SQL query to create list entities report	89
lib/database/mysql/db_mysql_list_jelines_report_sql.c	
Returns MYSQL SQL query to create JE lines report	90
lib/database/mysql/db_mysql_list_jes_report_sql.c	
Returns MYSQL SQL query to create journal entries report	90
lib/database/mysql/db_mysql_list_jesrcs_report_sql.c	
Returns MYSQL SQL query to create JE sources report	91
lib/database/mysql/db_mysql_list_nomaccts_report_sql.c	
Returns MYSQL SQL query to create list nominal accounts report	92
lib/database/mysql/db_mysql_list_users_report_sql.c	
Returns MYSQL SQL query to create list users report	92
lib/database/mysql/db_mysql_show_standingdata_report_sql.c	
Returns MYSQL SQL query to create show standing data report	93
lib/datastruct/data_structures.h	
Interface to data structures	94
lib/datastruct/ds_fieldtypes.h	
Record field types enumeration	94
lib/datastruct/ds_kvpair.h	
Implementation of string key-value pair data structure	95
lib/datastruct/ds_list.c	
Implementation of generic doubly-linked list data structure	98
lib/datastruct/ds_list.h	
Interface to generic doubly-linked list data structure	102
lib/datastruct/ds_map.c	
Implementation of string-string hash map data structure	106
lib/datastruct/ds_map.h	
Interface to string-string hash map data structure	109
lib/datastruct/ds_map_str.c	
Implementation of string-string hash map data structure	111

lib/datastruct/ ds_map_str.h	
Interface to string-string hash map data structure	113
lib/datastruct/ ds_record.c	
Implementation of record database structure	115
lib/datastruct/ ds_record.h	
Interface to record data structure	119
lib/datastruct/ ds_recordset.c	
Implementation of query result set structure	124
lib/datastruct/ ds_recordset.h	
Interface to record set structure	127
lib/datastruct/ ds_report.c	
Implementation of generic report data structure	132
lib/datastruct/ ds_report.h	
Interface to generic report data structure	134
lib/datastruct/ ds_str.c	
Implementation of string data structure	137
lib/datastruct/ ds_str.h	
Interface to string data structure	146
lib/datastruct/ ds_vector.c	
Implementation of generic doubly-linked vector data structure	156
lib/datastruct/ ds_vector.h	
Interface to generic doubly-linked vector data structure	160
lib/file_ops/ config_file_read.c	
Implementation of configuration file reading functionality	163
lib/file_ops/ config_file_read.h	
Interface to configuration file reading functionality	166
lib/file_ops/ delim_file_read.c	
Implementation of delimited file reading functionality	169
lib/file_ops/ delim_file_read.h	
Interface to delimited file reading functionality	171
lib/file_ops/ file_ops.h	
User interface to file operations functionality	172
lib/gl_general/ gl_config.c	
Implementation of configuration functionality	173
lib/gl_general/ gl_config.h	
Interface to configuration functionality	175
lib/gl_general/ gl_errors.c	
Implementation of error functionality	177
lib/gl_general/ gl_errors.h	
Interface to error functionality	178
lib/gl_general/ gl_general.h	
User interface to logging and error functionality	178
lib/gl_general/ gl_logging.c	
Implementation of logging functionality	179
lib/gl_general/ gl_logging.h	
Interface to logging functionality	181
lib/gl_general/ gl_login.c	
Implementation of login functionality	182
lib/gl_general/ gl_login.h	
Interface to login functionality	183
progs/gl_db/ gl_db_config.c	
Implementation of GL DB program configuration functionality	184
progs/gl_db/ gl_db_config.h	
Interface to GL DB program configuration functionality	186
progs/gl_db/ gl_db_main.c	
Main function for GL database program	188
progs/gl_reports/ gl_reports_config.c	
Implementation of GL reports program configuration functionality	189

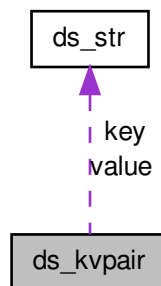
progs/gl_reports/ gl_reports_config.h	
Interface to GL reports program configuration functionality	191

Chapter 4

Data Structure Documentation

4.1 ds_kvpair Struct Reference

Collaboration diagram for ds_kvpair:



Data Fields

- [ds_str key](#)
- [ds_str value](#)

4.1.1 Detailed Description

Structure to hold a key-value pair node

4.1.2 Field Documentation

4.1.2.1 ds_str ds_kvpair::key

The key

4.1.2.2 `ds_str ds_kvpair::value`

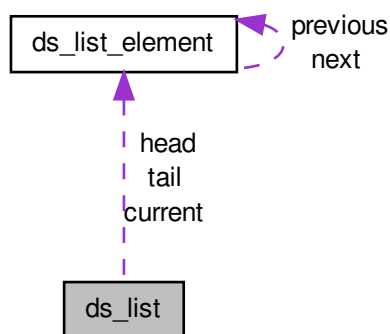
The value

The documentation for this struct was generated from the following file:

- `lib/datastruct/ds_kvpair.c`

4.2 `ds_list` Struct Reference

Collaboration diagram for `ds_list`:



Data Fields

- `size_t` [length](#)
- `bool` [free_on_delete](#)
- `struct ds_list_element *` [head](#)
- `struct ds_list_element *` [tail](#)
- `struct ds_list_element *` [current](#)
- `void(*` [data_destructor](#) `)(void *)`

4.2.1 Detailed Description

List data structure

4.2.2 Field Documentation

4.2.2.1 `struct ds_list_element* ds_list::current`

Pointer to current element

4.2.2.2 `void(* ds_list::data_destructor)(void *)`

Data destructor function

4.2.2.3 bool ds_list::free_on_delete

'Free on delete' flag

4.2.2.4 struct ds_list_element* ds_list::head

Pointer to head element

4.2.2.5 size_t ds_list::length

Length of list

4.2.2.6 struct ds_list_element* ds_list::tail

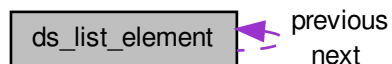
Pointer to tail element

The documentation for this struct was generated from the following file:

- lib/datastruct/[ds_list.c](#)

4.3 ds_list_element Struct Reference

Collaboration diagram for ds_list_element:



Data Fields

- void * [data](#)
- struct [ds_list_element](#) * [previous](#)
- struct [ds_list_element](#) * [next](#)

4.3.1 Detailed Description

List element data structure

4.3.2 Field Documentation

4.3.2.1 void* ds_list_element::data

Pointer to data

4.3.2.2 struct `ds_list_element*` `ds_list_element::next`

Pointer to next element

4.3.2.3 struct `ds_list_element*` `ds_list_element::previous`

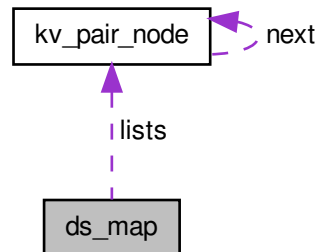
Pointer to previous element

The documentation for this struct was generated from the following file:

- [lib/datastruct/ds_list.c](#)

4.4 ds_map Struct Reference

Collaboration diagram for `ds_map`:



Data Fields

- struct `kv_pair_node` ** `lists`
- `size_t` `hash_size`

4.4.1 Detailed Description

Structure to hold a hash map

4.4.2 Field Documentation

4.4.2.1 `size_t` `ds_map::hash_size`

Size of array of lists

4.4.2.2 struct `kv_pair_node`** `ds_map::lists`

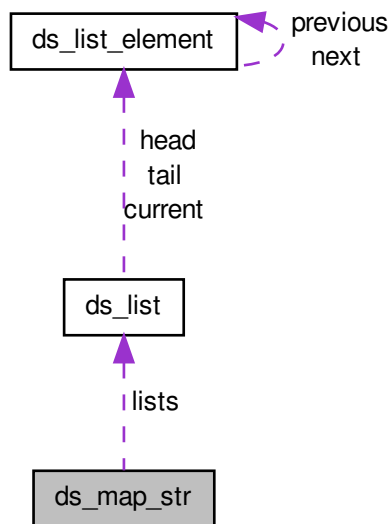
Pointer to array of lists

The documentation for this struct was generated from the following file:

- [lib/datastruct/ds_map.c](#)

4.5 ds_map_str Struct Reference

Collaboration diagram for ds_map_str:



Data Fields

- [ds_list * lists](#)
- `size_t` [hash_size](#)

4.5.1 Detailed Description

Structure to hold a hash map

4.5.2 Field Documentation

4.5.2.1 `size_t ds_map_str::hash_size`

Size of array of lists

4.5.2.2 `ds_list* ds_map_str::lists`

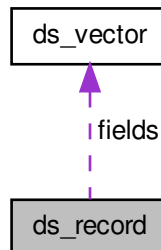
Array of lists

The documentation for this struct was generated from the following file:

- [lib/datastruct/ds_map_str.c](#)

4.6 ds_record Struct Reference

Collaboration diagram for ds_record:



Data Fields

- struct [ds_vector](#) * [fields](#)

4.6.1 Detailed Description

Vector data structure

4.6.2 Field Documentation

4.6.2.1 struct [ds_vector](#)* [ds_record::fields](#)

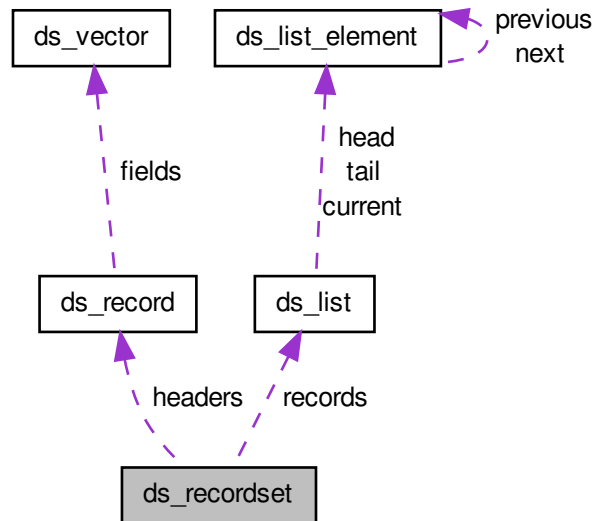
Vector of fields

The documentation for this struct was generated from the following file:

- lib/datastruct/[ds_record.c](#)

4.7 ds_recordset Struct Reference

Collaboration diagram for ds_recordset:



Data Fields

- `size_t num_fields`
- `size_t * field_lengths`
- `ds_record headers`
- `ds_list records`
- `enum ds_field_types * types`

4.7.1 Detailed Description

Result set structure

4.7.2 Field Documentation

4.7.2.1 `size_t* ds_recordset::field_lengths`

Lengths of the longest fields

4.7.2.2 `ds_record ds_recordset::headers`

A list of field headers

4.7.2.3 `size_t ds_recordset::num_fields`

The number of fields in a record

4.7.2.4 `ds_list` `ds_recordset::records`

A list of records

4.7.2.5 `enum ds_field_types*` `ds_recordset::types`

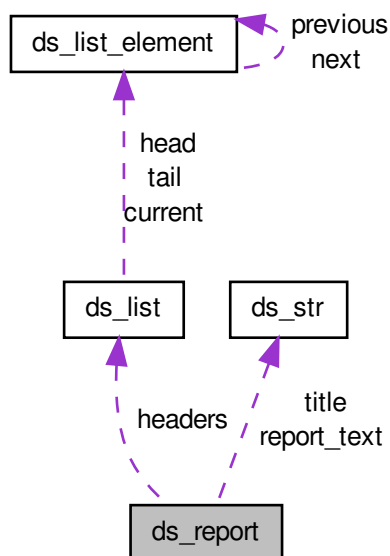
Types of records

The documentation for this struct was generated from the following file:

- [lib/datastruct/ds_recordset.c](#)

4.8 `ds_report` Struct Reference

Collaboration diagram for `ds_report`:



Data Fields

- [ds_str](#) `title`
- [ds_list](#) `headers`
- [ds_str](#) `report_text`
- [time_t](#) `created_time`

4.8.1 Detailed Description

Structure to contain a report

4.8.2 Field Documentation

4.8.2.1 time_t ds_report::created_time

The created time of the report

4.8.2.2 ds_list ds_report::headers

The report headers

4.8.2.3 ds_str ds_report::report_text

The report text

4.8.2.4 ds_str ds_report::title

The report title

The documentation for this struct was generated from the following file:

- lib/datastruct/[ds_report.c](#)

4.9 ds_str Struct Reference

Data Fields

- char * [data](#)
- size_t [length](#)
- size_t [capacity](#)

4.9.1 Detailed Description

Structure to contain string

4.9.2 Field Documentation

4.9.2.1 size_t ds_str::capacity

The size of the `data` buffer

4.9.2.2 char* ds_str::data

The data in C-style string format

4.9.2.3 size_t ds_str::length

The length of the string

The documentation for this struct was generated from the following file:

- lib/datastruct/[ds_str.c](#)

4.10 ds_vector Struct Reference

Data Fields

- `size_t` [size](#)
- `size_t` [current](#)
- `bool` [free_on_delete](#)
- `void **` [data](#)
- `void(* data_destructor)(void *)`

4.10.1 Detailed Description

Vector data structure

4.10.2 Field Documentation

4.10.2.1 `size_t ds_vector::current`

Current position

4.10.2.2 `void** ds_vector::data`

Data array

4.10.2.3 `void(* ds_vector::data_destructor)(void *)`

Data destructor function

4.10.2.4 `bool ds_vector::free_on_delete`

'Free on delete' flag

4.10.2.5 `size_t ds_vector::size`

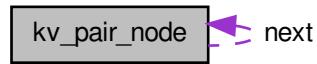
Size of vector

The documentation for this struct was generated from the following file:

- `lib/datastruct/ds_vector.c`

4.11 kv_pair_node Struct Reference

Collaboration diagram for kv_pair_node:



Data Fields

- char * [key](#)
- char * [value](#)
- struct [kv_pair_node](#) * [next](#)

4.11.1 Detailed Description

Structure to hold a key-value pair node

4.11.2 Field Documentation

4.11.2.1 char* kv_pair_node::key

A pointer to the key

4.11.2.2 struct kv_pair_node* kv_pair_node::next

A pointer to the next node

4.11.2.3 char* kv_pair_node::value

A pointer to the value

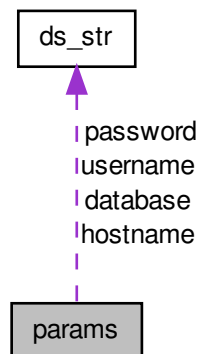
The documentation for this struct was generated from the following file:

- [lib/datastruct/ds_map.c](#)

4.12 params Struct Reference

```
#include <gl_config.h>
```

Collaboration diagram for params:



Data Fields

- [ds_str hostname](#)
- [ds_str database](#)
- [ds_str username](#)
- [ds_str password](#)

4.12.1 Detailed Description

Structure to hold database login parameters

4.12.2 Field Documentation

4.12.2.1 `ds_str params::database`

Database name

4.12.2.2 `ds_str params::hostname`

Database hostname

4.12.2.3 `ds_str params::password`

Password for database access

4.12.2.4 `ds_str params::username`

Username for database access

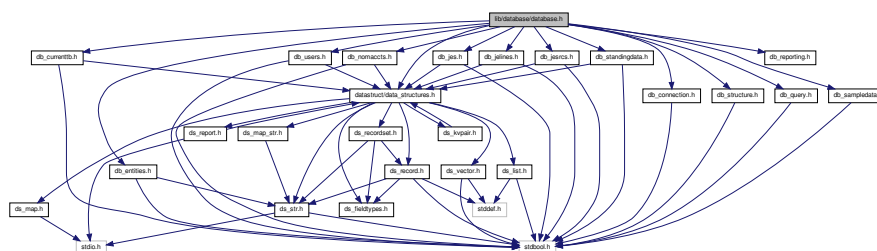
The documentation for this struct was generated from the following file:

- [lib/gl_general/gl_config.h](#)

File Documentation

User interface to database functionality.

Include dependency graph for database.h:

[illegible]

5.1.1 Detailed Description

User interface to database functionality.

Author

Paul Griffiths

Copyright

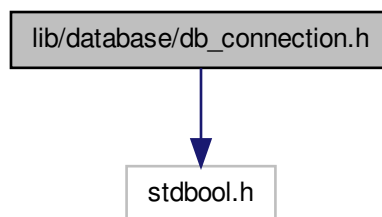
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.2 lib/database/db_connection.h File Reference

Interface to database connection functionality.

```
#include <stdbool.h>
```

Include dependency graph for db_connection.h:



This graph shows which files directly or indirectly include this file:



Functions

- bool **db_connect** (const char *host, const char *database, const char *username, const char *password)
Connects to a database.
- void **db_close** (void)
Disconnects from a database.

5.2.1 Detailed Description

Interface to database connection functionality. Function implementations are provided by the individual database components.

- `bool db_create_check_total_view (void)`
Creates the check total view in the database.
- `bool db_drop_check_total_view (void)`
Drops the check total view from the database.
- `ds_str db_check_total_report (ds_str entity)`
Runs the check total report.

5.3.1 Detailed Description

Implementation of current TB functionality.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.3.2 Function Documentation

5.3.2.1 `ds_str db_check_total_report (ds_str entity)`

Runs the check total report.

Returns

The report.

5.3.2.2 `bool db_create_check_total_view (void)`

Creates the check total view in the database.

Returns

`true` on success, `false` on failure.

5.3.2.3 `bool db_create_current_trial_balance_view (void)`

Creates the current TB view in the database.

Returns

`true` on success, `false` on failure.

5.3.2.4 `ds_str db_current_trial_balance_report (ds_str entity)`

Runs the current trial balance report.

Returns

The report.

- `ds_str db_current_trial_balance_report (ds_str entity)`
Runs the current trial balance report.
- `bool db_create_check_total_view (void)`
Creates the check total view in the database.
- `bool db_drop_check_total_view (void)`
Drops the check total view from the database.
- `ds_str db_check_total_report (ds_str entity)`
Runs the check total report.

5.4.1 Detailed Description

Interface to current trial balance functionality.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.4.2 Function Documentation

5.4.2.1 `ds_str db_check_total_report (ds_str entity)`

Runs the check total report.

Returns

The report.

5.4.2.2 `bool db_create_check_total_view (void)`

Creates the check total view in the database.

Returns

`true` on success, `false` on failure.

5.4.2.3 `bool db_create_current_trial_balance_view (void)`

Creates the current TB view in the database.

Returns

`true` on success, `false` on failure.

5.4.2.4 `ds_str db_current_trial_balance_report (ds_str entity)`

Runs the current trial balance report.

Returns

The report.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.5.2 Function Documentation**5.5.2.1 `bool db_create_entities_table (void)`**

Creates the entities table in the database.

Returns

`true` on success, `false` on failure.

5.5.2.2 `bool db_drop_entities_table (void)`

Drops the entities table in the database.

Returns

`true` on success, `false` on failure.

5.5.2.3 `ds_str db_get_entity_name_from_id (ds_str entity_id)`

Returns an entity name from an ID.

Parameters

<i>entity_id</i>	The entity ID.
------------------	----------------

Returns

The string, containing an "Unknown entity" string if the ID is not found.

5.5.2.4 `ds_str db_list_entities_report (void)`

Creates a report listing all entities.

Returns

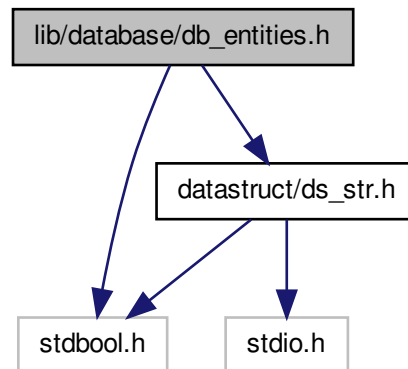
A `ds_str` containing the report.

5.6 `lib/database/db_entities.h` File Reference

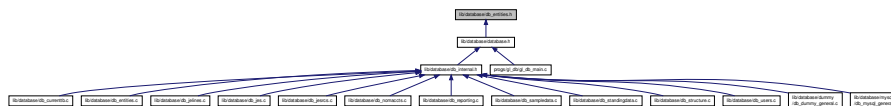
Interface to entities functionality.

```
#include <stdbool.h>
#include "datastruct/ds_str.h"
```

Include dependency graph for db_entities.h:



This graph shows which files directly or indirectly include this file:



Functions

- bool `db_create_entities_table` (void)
Creates the entities table in the database.
- bool `db_drop_entities_table` (void)
Drops the entities table in the database.
- `ds_str` `db_list_entities_report` (void)
Creates a report listing all entities.
- `ds_str` `db_get_entity_name_from_id` (`ds_str` entity_id)
Returns an entity name from an ID.

5.6.1 Detailed Description

Interface to entities functionality.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.6.2 Function Documentation

5.6.2.1 `bool db.create_entities_table (void)`

Creates the entities table in the database.

Returns

`true` on success, `false` on failure.

5.6.2.2 `bool db.drop_entities_table (void)`

Drops the entities table in the database.

Returns

`true` on success, `false` on failure.

5.6.2.3 `ds_str db.get_entity_name_from_id (ds_str entity_id)`

Returns an entity name from an ID.

Parameters

<code>entity_id</code>	The entity ID.
------------------------	----------------

Returns

The string, containing an "Unknown entity" string if the ID is not found.

5.6.2.4 `ds_str db.list_entities_report (void)`

Creates a report listing all entities.

5.8.2.3 ds_str db_list_jelines_report (void)

Creates a report listing all journal entry lines..

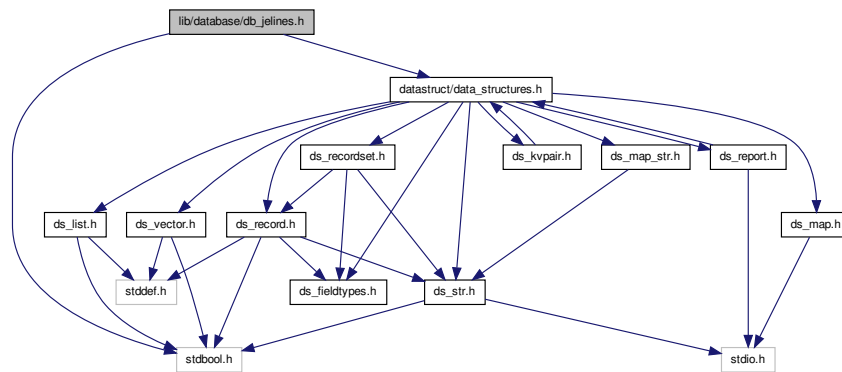
Returns

A `ds_str` containing the report.

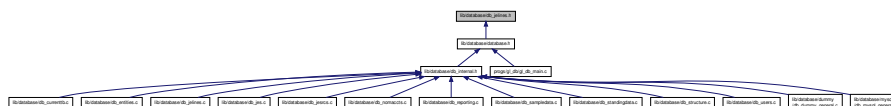
5.9 lib/database/db_jelines.h File Reference

Interface to journal entry lines functionality.

```
#include <stdbool.h>
#include "datastruct/data_structures.h"
Include dependency graph for db_jelines.h:
```



This graph shows which files directly or indirectly include this file:



Functions

- bool `db_create_jelines_table` (void)
Creates the journal entry lines table in the database.
- bool `db_drop_jelines_table` (void)
Drops the journal entry lines table from the database.
- `ds_str db_list_jelines_report` (void)
Creates a report listing all journal entry lines..

5.9.1 Detailed Description

Interface to journal entry lines functionality.

Functions

- bool [db_create_jes_table](#) (void)
Creates the journal entries table in the database.
- bool [db_drop_jes_table](#) (void)
Drops the jes table from the database.
- [ds_str db_list_jes_report](#) (void)
Creates a report listing all journal entries.
- bool [db_create_all_jes_view](#) (void)
Creates the all JEs view in the database.
- bool [db_drop_all_jes_view](#) (void)
Drops the all JEs view from the database.
- [ds_str db_all_jes_report](#) ([ds_str](#) je_num)
Creates a report showing all journal entries.

5.10.1 Detailed Description

Implementation of journal entries functionality.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.10.2 Function Documentation

5.10.2.1 [ds_str db_all_jes_report](#) ([ds_str](#) je_num)

Creates a report showing all journal entries.

Parameters

je_num	The journal entry number to show, or <code>NULL</code> to show all journal entries.
------------------------	---

Returns

A [ds_str](#) containing the report.

5.10.2.2 [bool db_create_all_jes_view](#) (void)

Creates the all JEs view in the database.

Returns

`true` on success, `false` on failure.

5.10.2.3 [bool db_create_jes_table](#) (void)

Creates the journal entries table in the database.

Returns

`true` on success, `false` on failure.

5.10.2.4 `bool db_drop_all_jes_view (void)`

Drops the all JEs view from the database.

Returns

`true` on success, `false` on failure.

5.10.2.5 `bool db_drop_jes_table (void)`

Drops the jes table from the database.

Returns

`true` on success, `false` on failure.

5.10.2.6 `ds_str db_list_jes_report (void)`

Creates a report listing all journal entries.

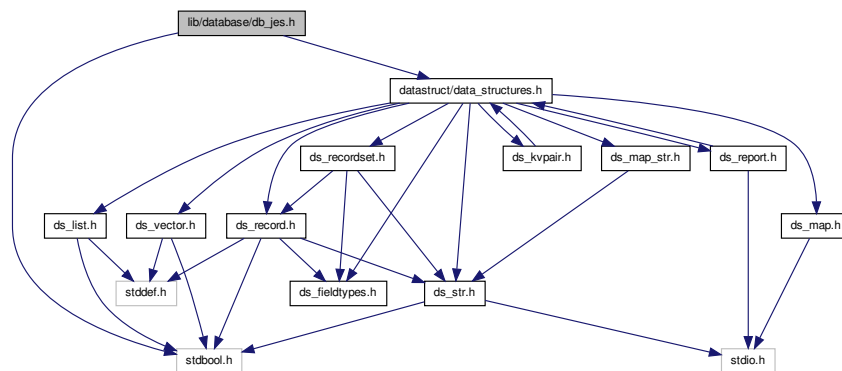
Returns

A `ds_str` containing the report.

5.11 `lib/database/db_jes.h` File Reference

Interface to journal entries functionality.

```
#include <stdbool.h>
#include "datastruct/data_structures.h"
Include dependency graph for db_jes.h:
```



[illegible]

- `bool db_create_jes_table (void)`
Creates the journal entries table in the database.
- `bool db_drop_jes_table (void)`
Drops the jes table from the database.
- `ds_str db_list_jes_report (void)`
Creates a report listing all journal entries.
- `bool db_create_all_jes_view (void)`
Creates the all JEs view in the database.
- `bool db_drop_all_jes_view (void)`
Drops the all JEs view from the database.
- `ds_str db_all_jes_report (ds_str je_num)`
Creates a report showing all journal entries.

Author
Paul Griffiths

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

<i>je_num</i>	The journal entry number to show, or <code>NULL</code> to show all journal entries.
---------------	---

Creates the all JEs view in the database.

Functions

- `bool db_create_jesrcs_table` (void)
Creates the JE sources table in the database.
- `bool db_drop_jesrcs_table` (void)
Drops the jesrcs table from the database.
- `ds_str db_list_jesrcs_report` (void)
Creates a report listing all journal entry sources.

5.12.1 Detailed Description

Implementation of journal entry sources functionality.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.12.2 Function Documentation

5.12.2.1 `bool db_create_jesrcs_table (void)`

Creates the JE sources table in the database.

Returns

`true` on success, `false` on failure.

5.12.2.2 `bool db_drop_jesrcs_table (void)`

Drops the jesrcs table from the database.

Returns

`true` on success, `false` on failure.

5.12.2.3 `ds_str db_list_jesrcs_report (void)`

Creates a report listing all journal entry sources.

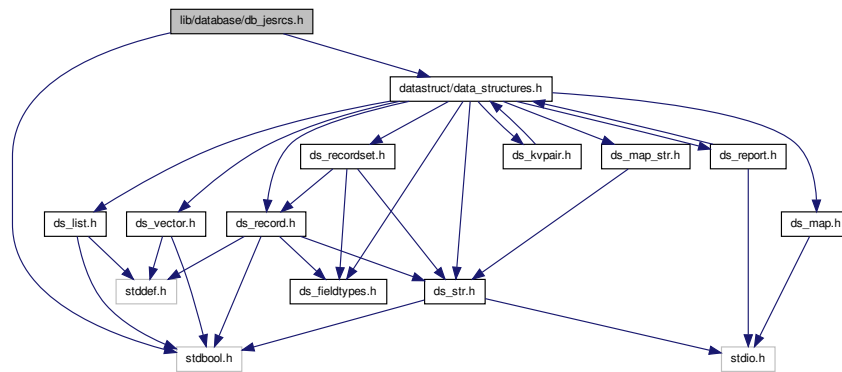
Returns

A `ds_str` containing the report.

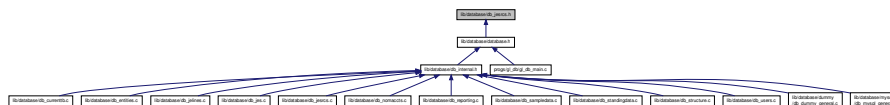
5.13 lib/database/db_jesrcs.h File Reference

Interface to journal entry sources functionality.

```
#include <stdbool.h>
#include "datastruct/data_structures.h"
Include dependency graph for db_jesrcs.h:
```



This graph shows which files directly or indirectly include this file:



Functions

- bool [db_create_jesrcs_table](#) (void)
Creates the JE sources table in the database.
- bool [db_drop_jesrcs_table](#) (void)
Drops the jesrcs table from the database.
- [ds_str db_list_jesrcs_report](#) (void)
Creates a report listing all journal entry sources.

5.13.1 Detailed Description

Interface to journal entry sources functionality.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.14.1 Detailed Description

Implementation of nominal accounts functionality.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.14.2 Function Documentation

5.14.2.1 `bool db_create_nomaccts_table (void)`

Creates the nominal accounts table in the database.

Returns

`true` on success, `false` on failure.

5.14.2.2 `bool db_drop_nomaccts_table (void)`

Drops the nomaccts table from the database.

Returns

`true` on success, `false` on failure.

5.14.2.3 `ds_str db_list_nomaccts_report (void)`

Creates a report listing all nominal accounts.

Returns

A `ds_str` containing the report.

5.15 `lib/database/db_nomaccts.h` File Reference

Interface to nominal accounts functionality.

```
#include <stdbool.h>
#include "datastruct/data_structures.h"
```

```

graph TD
    lib["lib/database/db_nomaccts.h"] --> datastruct["datastruct/data_structures.h"]
    datastruct --> ds_recordset["ds_recordset.h"]
    datastruct --> ds_kvpair["ds_kvpair.h"]
    datastruct --> ds_map_str["ds_map_str.h"]
    datastruct --> ds_report["ds_report.h"]
    datastruct --> ds_map["ds_map.h"]
    datastruct --> ds_str["ds_str.h"]
    datastruct --> ds_fieldtypes["ds_fieldtypes.h"]
    datastruct --> ds_record["ds_record.h"]
    datastruct --> ds_vector["ds_vector.h"]
    datastruct --> ds_list["ds_list.h"]
    ds_recordset --> ds_record
    ds_recordset --> ds_str
    ds_recordset --> ds_fieldtypes
    ds_kvpair --> ds_str
    ds_kvpair --> ds_fieldtypes
    ds_map_str --> ds_str
    ds_map_str --> ds_fieldtypes
    ds_report --> ds_str
    ds_report --> ds_fieldtypes
    ds_map --> ds_str
    ds_map --> ds_fieldtypes
    ds_str --> ds_fieldtypes
    ds_record --> ds_str
    ds_record --> ds_fieldtypes
    ds_vector --> ds_str
    ds_vector --> ds_fieldtypes
    ds_list --> ds_str
    ds_list --> ds_fieldtypes
    ds_str --> ds_fieldtypes
    ds_fieldtypes --> stdbool["stdbool.h"]
    ds_fieldtypes --> stdio["stdio.h"]
    stdbool --> stdio

```

- bool `db_create_nomaccts_table` (void)
Creates the nominal accounts table in the database.
- bool `db_drop_nomaccts_table` (void)
Drops the nomaccts table from the database.
- `ds_str db_list_nomaccts_report` (void)
Creates a report listing all nominal accounts.

Paul Griffiths

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

Creates the nominal accounts table in the database.

Returns

`true` on success, `false` on failure.

5.15.2.2 `bool db_drop_nomaccts_table (void)`

Drops the nomaccts table from the database.

Returns

`true` on success, `false` on failure.

5.15.2.3 `ds_str db_list_nomaccts_report (void)`

Creates a report listing all nominal accounts.

Returns

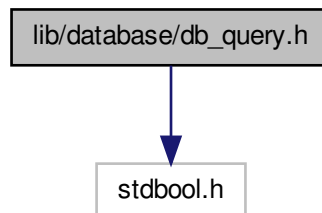
A `ds_str` containing the report.

5.16 `lib/database/db_query.h` File Reference

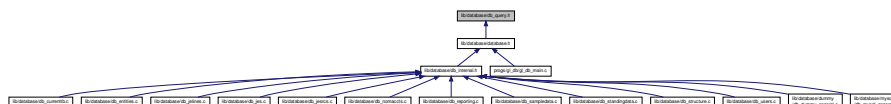
Interface to database query functionality.

```
#include <stdbool.h>
```

Include dependency graph for `db_query.h`:



This graph shows which files directly or indirectly include this file:

**Functions**

- `bool db_execute_query (ds_str query)`
Executes an SQL query on the database.

5.17.1 Detailed Description

Implementation of database reporting functionality.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.17.2 Function Documentation

5.17.2.1 `ds_str db_create_report_from_query (ds_str query)`

Creates a text report from a query.

Parameters

<code>query</code>	The SELECT query to run.
--------------------	--------------------------

Returns

A `ds_str` containing the report, or `NULL` on failure.

5.18 lib/database/db_reporting.h File Reference

Interface to database reporting functionality.

This graph shows which files directly or indirectly include this file:



Functions

- `ds_str db_create_report_from_query (ds_str query)`
Creates a text report from a query.
- `ds_recordset db_create_recordset_from_query (ds_str query)`
Creates a `ds_recordset` from a query.

5.18.1 Detailed Description

Interface to database reporting functionality. Function implementations may be provided by the individual database components.

Author

Paul Griffiths

Functions

- bool `db_load_sample_data` (void)
Loads sample data into the database.

5.19.1 Detailed Description

Implementation of database sample data functionality.

Author

Paul Griffiths

Copyright

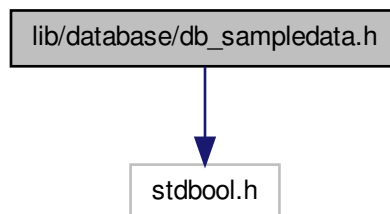
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.20 lib/database/db_sampledata.h File Reference

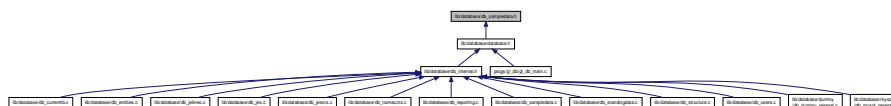
Interface to database sample data functionality.

```
#include <stdbool.h>
```

Include dependency graph for db_sampledata.h:



This graph shows which files directly or indirectly include this file:



Functions

- bool `db_load_sample_data` (void)
Loads sample data into the database.

5.20.1 Detailed Description

Interface to database sample data functionality.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.21 lib/database/db_sql.h File Reference

Interface to database specific SQL strings.

This graph shows which files directly or indirectly include this file:



Functions

- const char * [db_create_users_table_sql](#) (void)
Returns the SQL query to create the users table.
- const char * [db_drop_users_table_sql](#) (void)
Returns the SQL query to drop the users table.
- const char * [db_list_users_report_sql](#) (void)
Returns the SQL query to run the "list users" report.
- const char * [db_create_entities_table_sql](#) (void)
Returns the SQL query to create the entities table.
- const char * [db_drop_entities_table_sql](#) (void)
Returns the SQL query to drop the entities table.
- const char * [db_list_entities_report_sql](#) (void)
Returns the SQL query to run the "list entities" report.
- const char * [db_create_jes_table_sql](#) (void)
Returns the SQL query to create the journal entries table.
- const char * [db_drop_jes_table_sql](#) (void)
Returns the SQL query to drop the journal entries table.
- const char * [db_list_jes_report_sql](#) (void)
Returns the SQL query to run the "list journal entries" report.
- const char * [db_create_nomaccts_table_sql](#) (void)
Returns the SQL query to create the nominal accounts table.
- const char * [db_drop_nomaccts_table_sql](#) (void)
Returns the SQL query to drop the nominal accounts table.
- const char * [db_list_nomaccts_report_sql](#) (void)
Returns the SQL query to run the "list nominal accounts" report.
- const char * [db_create_jelines_table_sql](#) (void)
Returns the SQL query to create the JE lines table.

- const char * [db_drop_jelines_table_sql](#) (void)
Returns the SQL query to drop the JE lines table.
- const char * [db_list_jelines_report_sql](#) (void)
Returns the SQL query to run the "list JE lines" report.
- const char * [db_current_trial_balance_report_sql](#) (void)
Returns the SQL query to run the "current TB" report.
- const char * [db_current_trial_balance_entity_report_sql](#) (void)
Returns the SQL query to run the "current TB" by entity. report.
- const char * [db_create_jesrcs_table_sql](#) (void)
Returns the SQL query to create the JE sources table.
- const char * [db_drop_jesrcs_table_sql](#) (void)
Returns the SQL query to drop the JE sources table.
- const char * [db_list_jesrcs_report_sql](#) (void)
Returns the SQL query to run the "list JE sources" report.
- const char * [db_create_standingdata_table_sql](#) (void)
Returns the SQL query to create the standing data table.
- const char * [db_drop_standingdata_table_sql](#) (void)
Returns the SQL query to drop the standing data table.
- const char * [db_show_standingdata_report_sql](#) (void)
Returns the SQL query to run the "show standing data" report.
- const char * [db_create_current_trial_balance_view_sql](#) (void)
Returns the SQL query to create the current TB view.
- const char * [db_drop_current_trial_balance_view_sql](#) (void)
Returns the SQL query to drop the current TB view.
- const char * [db_create_check_total_view_sql](#) (void)
Returns the SQL query to create the check total view.
- const char * [db_drop_check_total_view_sql](#) (void)
Returns the SQL query to drop the check total view.
- const char * [db_check_total_report_sql](#) (void)
Returns the SQL query to run the "check total" report.
- const char * [db_check_total_entity_report_sql](#) (void)
Returns the SQL query to run the "check total" by entity. report.
- const char * [db_create_all_jes_view_sql](#) (void)
Returns the SQL query to create the all JEs view.
- const char * [db_drop_all_jes_view_sql](#) (void)
Returns the SQL query to drop the all JEs view.
- const char * [db_all_jes_report_sql](#) (void)
Returns the SQL query to run the "all JEs" report.
- const char * [db_all_jes_number_report_sql](#) (void)
Returns the SQL query to run the "JE by number" report.
- const char * [db_get_entity_name_from_id_sql](#) (void)
Returns the SQL query to get an entity name from its ID.

5.21.1 Detailed Description

Interface to database specific SQL strings. Function implementations are provided by the individual database components.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.21.2 Function Documentation

5.21.2.1 `const char* db_all_jes_number_report_sql (void)`

Returns the SQL query to run the "JE by number" report.

Returns

The SQL query.

5.21.2.2 `const char* db_all_jes_report_sql (void)`

Returns the SQL query to run the "all JEs" report.

Returns

The SQL query.

5.21.2.3 `const char* db_check_total_entity_report_sql (void)`

Returns the SQL query to run the "check total" by entity. report.

Returns

The SQL query.

5.21.2.4 `const char* db_check_total_report_sql (void)`

Returns the SQL query to run the "check total" report.

Returns

The SQL query.

5.21.2.5 `const char* db_create_all_jes_view_sql (void)`

Returns the SQL query to create the all JEs view.

Returns

The SQL query.

5.21.2.6 `const char* db_create_check_total_view_sql (void)`

Returns the SQL query to create the check total view.

Returns

The SQL query.

5.21.2.7 const char* db_create_current_trial_balance_view_sql (void)

Returns the SQL query to create the current TB view.

Returns

The SQL query.

5.21.2.8 const char* db_create_entities_table_sql (void)

Returns the SQL query to create the entities table.

Returns

The SQL query.

5.21.2.9 const char* db_create_jelines_table_sql (void)

Returns the SQL query to create the JE lines table.

Returns

The SQL query.

5.21.2.10 const char* db_create_jes_table_sql (void)

Returns the SQL query to create the journal entries table.

Returns

The SQL query.

5.21.2.11 const char* db_create_jesrcs_table_sql (void)

Returns the SQL query to create the JE sources table.

Returns

The SQL query.

5.21.2.12 const char* db_create_nomaccts_table_sql (void)

Returns the SQL query to create the nominal accounts table.

Returns

The SQL query.

5.21.2.13 const char* db_create_standingdata_table_sql (void)

Returns the SQL query to create the standing data table.

Returns

The SQL query.

5.21.2.14 const char* db_create_users_table_sql (void)

Returns the SQL query to create the users table.

Returns

The SQL query.

5.21.2.15 const char* db_current_trial_balance_entity_report_sql (void)

Returns the SQL query to run the "current TB" by entity. report.

Returns

The SQL query.

5.21.2.16 const char* db_current_trial_balance_report_sql (void)

Returns the SQL query to run the "current TB" report.

Returns

The SQL query.

5.21.2.17 const char* db_drop_all_jes_view_sql (void)

Returns the SQL query to drop the all JEs view.

Returns

The SQL query.

5.21.2.18 const char* db_drop_check_total_view_sql (void)

Returns the SQL query to drop the check total view.

Returns

The SQL query.

5.21.2.19 const char* db_drop_current_trial_balance_view_sql (void)

Returns the SQL query to drop the current TB view.

Returns

The SQL query.

5.21.2.20 const char* db_drop_entities_table_sql (void)

Returns the SQL query to drop the entities table.

Returns

The SQL query.

5.21.2.21 `const char* db_drop_jelines_table_sql (void)`

Returns the SQL query to drop the JE lines table.

Returns

The SQL query.

5.21.2.22 `const char* db_drop_jes_table_sql (void)`

Returns the SQL query to drop the journal entries table.

Returns

The SQL query.

5.21.2.23 `const char* db_drop_jesrcs_table_sql (void)`

Returns the SQL query to drop the JE sources table.

Returns

The SQL query.

5.21.2.24 `const char* db_drop_nomaccts_table_sql (void)`

Returns the SQL query to drop the nominal accounts table.

Returns

The SQL query.

5.21.2.25 `const char* db_drop_standingdata_table_sql (void)`

Returns the SQL query to drop the standing data table.

Returns

The SQL query.

5.21.2.26 `const char* db_drop_users_table_sql (void)`

Returns the SQL query to drop the users table.

Returns

The SQL query.

5.21.2.27 `const char* db_get_entity_name_from_id_sql (void)`

Returns the SQL query to get an entity name from its ID.

\

Returns

The SQL query.

5.21.2.28 `const char* db_list_entities_report_sql (void)`

Returns the SQL query to run the "list entities" report.

Returns

The SQL query.

5.21.2.29 `const char* db_list_jelines_report_sql (void)`

Returns the SQL query to run the "list JE lines" report.

Returns

The SQL query.

5.21.2.30 `const char* db_list_jes_report_sql (void)`

Returns the SQL query to run the "list journal entries" report.

Returns

The SQL query.

5.21.2.31 `const char* db_list_jesrcs_report_sql (void)`

Returns the SQL query to run the "list JE sources" report.

Returns

The SQL query.

5.21.2.32 `const char* db_list_nomaccts_report_sql (void)`

Returns the SQL query to run the "list nominal accounts" report.

Returns

The SQL query.

5.21.2.33 `const char* db_list_users_report_sql (void)`

Returns the SQL query to run the "list users" report.

Returns

The SQL query.

5.21.2.34 `const char* db_show_standingdata_report_sql (void)`

Returns the SQL query to run the "show standing data" report.

Returns

The SQL query.

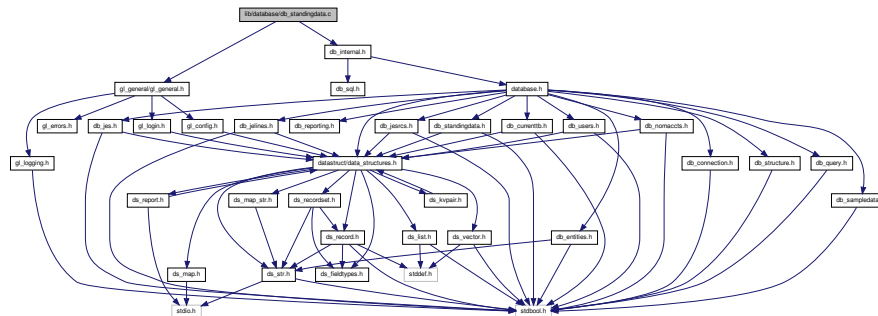
5.22 lib/database/db_standingdata.c File Reference

Implementation of standing data functionality.

```
#include "gl_general/gl_general.h"
```

```
#include "db_internal.h"
```

Include dependency graph for db_standingdata.c:



Functions

- `bool db_create_standingdata_table (void)`
Creates the standing data table in the database.
- `bool db_drop_standingdata_table (void)`
Drops the standingdata table from the database.
- `ds_str db_show_standingdata_report (void)`
Creates a report showing standing data.

5.22.1 Detailed Description

Implementation of standing data functionality.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.22.2 Function Documentation

5.22.2.1 `bool db_create_standingdata_table (void)`

Creates the standing data table in the database.

Returns

`true` on success, `false` on failure.

5.22.2.2 bool db_drop_standingdata_table (void)

Drops the standingdata table from the database.

Returns

true on success, false on failure.

5.22.2.3 ds_str db_show_standingdata_report (void)

Creates a report showing standing data.

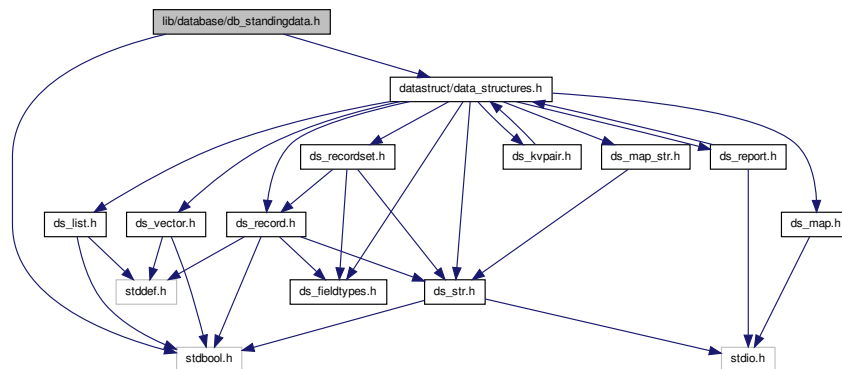
Returns

A [ds_str](#) containing the report.

5.23 lib/database/db_standingdata.h File Reference

Interface to journal entries functionality.

```
#include <stdbool.h>
#include "datastruct/data_structures.h"
Include dependency graph for db_standingdata.h:
```



This graph shows which files directly or indirectly include this file:



Functions

- bool [db_create_standingdata_table](#) (void)
Creates the standing data table in the database.
- bool [db_drop_standingdata_table](#) (void)
Drops the standingdata table from the database.

- [ds_str db_show_standingdata_report](#) (void)

Creates a report showing standing data.

5.23.1 Detailed Description

Interface to journal entries functionality.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.23.2 Function Documentation

5.23.2.1 `bool db_create_standingdata_table (void)`

Creates the standing data table in the database.

Returns

`true` on success, `false` on failure.

5.23.2.2 `bool db_drop_standingdata_table (void)`

Drops the standingdata table from the database.

Returns

`true` on success, `false` on failure.

5.23.2.3 `ds_str db_show_standingdata_report (void)`

Creates a report showing standing data.

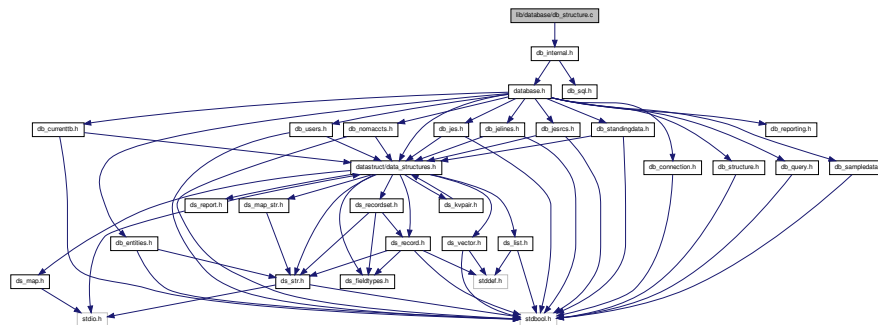
Returns

A [ds_str](#) containing the report.

5.24 lib/database/db_structure.c File Reference

Implementation of database structure functionality.

Include dependency graph for db_structure.c:



- bool `db_create_database_structure` (void)
Creates an empty database structure.
- bool `db_delete_database_structure` (void)
Deletes the database structure.

Implementation of database structure functionality.

Paul Griffiths

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.24.2.1 bool db_create_database_structure (void)

Returns

true on success, false on failure.

Deletes the database structure.

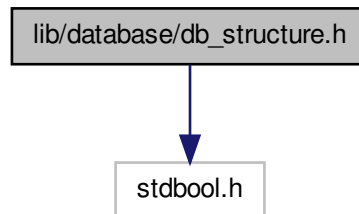
true on success, false on failure.

5.25 lib/database/db_structure.h File Reference

Interface to database structure functionality.

```
#include <stdbool.h>
```

Include dependency graph for db_structure.h:



This graph shows which files directly or indirectly include this file:



Functions

- bool [db_create_database_structure](#) (void)
Creates an empty database structure.
- bool [db_delete_database_structure](#) (void)
Deletes the database structure.

5.25.1 Detailed Description

Interface to database structure functionality.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.25.2 Function Documentation

5.25.2.1 bool db_create_database_structure (void)

Creates an empty database structure.

Returns

true on success, false on failure.

5.25.2.2 bool db_delete_database_structure (void)

Deletes the database structure.

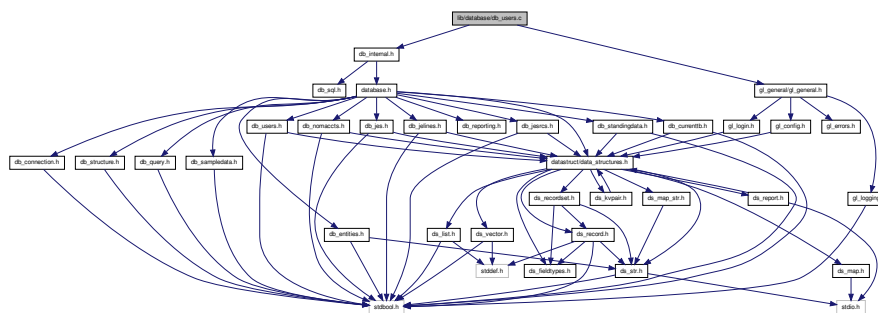
Returns

true on success, false on failure.

5.26 lib/database/db_users.c File Reference

Implementation of users functionality.

```
#include "db_internal.h"
#include "gl_general/gl_general.h"
Include dependency graph for db_users.c:
```



Functions

- bool `db_create_users_table` (void)
Creates the users table in the database.
- bool `db_drop_users_table` (void)
Drops the users table from the database.
- `ds_str` `db_list_users_report` (void)
Creates a report listing all users.

5.26.1 Detailed Description

Implementation of users functionality.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.26.2 Function Documentation

5.26.2.1 `bool db_create_users_table (void)`

Creates the users table in the database.

Returns

`true` on success, `false` on failure.

5.26.2.2 `bool db_drop_users_table (void)`

Drops the users table from the database.

Returns

`true` on success, `false` on failure.

5.26.2.3 `ds_str db_list_users_report (void)`

Creates a report listing all users.

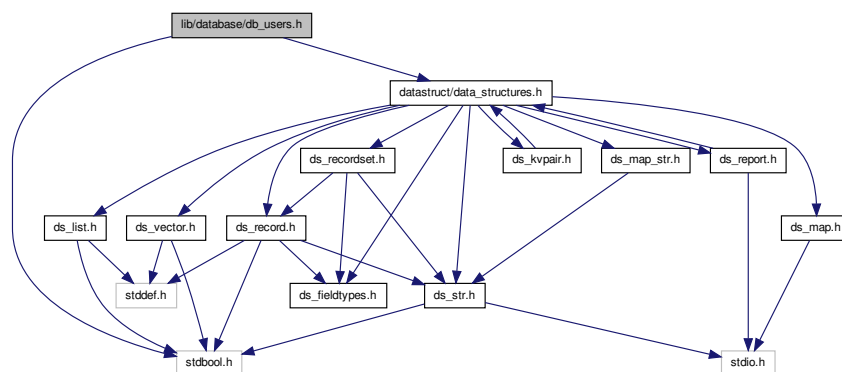
Returns

A `ds_str` containing the report.

5.27 `lib/database/db_users.h` File Reference

Interface to users functionality.

```
#include <stdbool.h>
#include "datastruct/data_structures.h"
Include dependency graph for db_users.h:
```



This graph shows which files directly or indirectly include this file:



Functions

- bool [db_create_users_table](#) (void)
Creates the users table in the database.
- bool [db_drop_users_table](#) (void)
Drops the users table from the database.
- [ds_str](#) [db_list_users_report](#) (void)
Creates a report listing all users.

5.27.1 Detailed Description

Interface to users functionality.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.27.2 Function Documentation

5.27.2.1 bool db_create_users_table (void)

Creates the users table in the database.

Returns

`true` on success, `false` on failure.

5.27.2.2 bool db_drop_users_table (void)

Drops the users table from the database.

Returns

`true` on success, `false` on failure.

5.27.2.3 ds_str db_list_users_report (void)

Creates a report listing all users.

Returns

A [ds_str](#) containing the report.

5.28 lib/database/dummy/db_dummy_create_entities_table_sql.c File Reference

Returns dummy SQL query to create entities table.

Functions

- `const char * db_create_entities_table_sql` (void)
Returns the SQL query to create the entities table.

5.28.1 Detailed Description

Returns dummy SQL query to create entities table.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.28.2 Function Documentation

5.28.2.1 `const char* db_create_entities_table_sql (void)`

Returns the SQL query to create the entities table.

Returns

The SQL query.

5.29 lib/database/dummy/db_dummy_create_users_table_sql.c File Reference

Returns dummy SQL query to create users table.

Functions

- `const char * db_create_users_table_sql` (void)
Returns the SQL query to create the users table.

5.29.1 Detailed Description

Returns dummy SQL query to create users table.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.29.2 Function Documentation

5.29.2.1 `const char* db_create_users_table_sql (void)`

Returns the SQL query to create the users table.

Returns

The SQL query.

5.30 lib/database/dummy/db_dummy_drop_entities_table_sql.c File Reference

Returns dummy SQL query to drop entities table.

Functions

- `const char * db_drop_entities_table_sql (void)`
Returns the SQL query to drop the entities table.

5.30.1 Detailed Description

Returns dummy SQL query to drop entities table.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.30.2 Function Documentation

5.30.2.1 `const char* db_drop_entities_table_sql (void)`

Returns the SQL query to drop the entities table.

Returns

The SQL query.

5.31 lib/database/dummy/db_dummy_drop_users_table_sql.c File Reference

Returns dummy SQL query to drop users table.

Functions

- `const char * db_drop_users_table_sql (void)`
Returns the SQL query to drop the users table.

Functions

- bool [db_connect](#) (const char *host, const char *database, const char *username, const char *password)
Connects to a database.
- void [db_close](#) (void)
Disconnects from a database.
- bool [db_execute_query](#) (ds_str query)
Executes an SQL query on the database.
- ds_recordset [db_create_recordset_from_query](#) (ds_str query)
Creates a [ds_recordset](#) from a query.

5.32.1 Detailed Description

Implementation of dummy database functionality. This module is useful when compiling for testing purpose on a system without any of the supported database development libraries available.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.32.2 Macro Definition Documentation

5.32.2.1 #define _XOPEN_SOURCE 600

UNIX feature test macro

5.32.3 Function Documentation

5.32.3.1 bool db_connect (const char * host, const char * database, const char * username, const char * password)

Connects to a database.

Parameters

<i>host</i>	The hostname.
<i>database</i>	The database name.
<i>username</i>	The username with which to connect.
<i>password</i>	The password for the specified user.

Returns

`true` if the connection was successfully made, `false` otherwise.

5.32.3.2 ds_recordset db_create_recordset_from_query (ds_str query)

Creates a [ds_recordset](#) from a query.

Parameters

<i>query</i>	The SELECT query to run.
--------------	--------------------------

Returns

A [ds_recordset](#) containing the query result, or `NULL` on failure.

5.32.3.3 bool db_execute_query (ds_str query)

Executes an SQL query on the database.

Parameters

<i>query</i>	The query to execute.
--------------	-----------------------

Returns

`true` if the query was successfully executed, `false` otherwise.

5.33 lib/database/dummy/db_dummy_list_entities_report_sql.c File Reference

Returns dummy SQL query to create list entities report.

Functions

- `const char * db_list_entities_report_sql (void)`
Returns the SQL query to run the "list entities" report.

5.33.1 Detailed Description

Returns dummy SQL query to create list entities report.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.33.2 Function Documentation**5.33.2.1 const char* db_list_entities_report_sql (void)**

Returns the SQL query to run the "list entities" report.

Returns

The SQL query.

5.34 lib/database/dummy/db_dummy_list_users_report_sql.c File Reference

Returns dummy SQL query to create list users report.

Functions

- const char * [db_list_users_report_sql](#) (void)
Returns the SQL query to run the "list users" report.

5.34.1 Detailed Description

Returns dummy SQL query to create list users report.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.34.2 Function Documentation

5.34.2.1 const char* db_list_users_report_sql (void)

Returns the SQL query to run the "list users" report.

Returns

The SQL query.

5.35 lib/database/mysql/db_mysql_all_jes_number_report_sql.c File Reference

Returns MYSQL SQL query to create JE by number report.

Functions

- const char * [db_all_jes_number_report_sql](#) (void)
Returns the SQL query to run the "JE by number" report.

5.35.1 Detailed Description

Returns MYSQL SQL query to create JE by number report.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.35.2 Function Documentation

5.35.2.1 `const char* db_all_jes_number_report_sql (void)`

Returns the SQL query to run the "JE by number" report.

Returns

The SQL query.

5.36 `lib/database/mysql/db_mysql_all_jes_report_sql.c` File Reference

Returns MYSQL SQL query to create all jes report.

Functions

- `const char * db_all_jes_report_sql (void)`
Returns the SQL query to run the "all JEs" report.

5.36.1 Detailed Description

Returns MYSQL SQL query to create all jes report.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.36.2 Function Documentation

5.36.2.1 `const char* db_all_jes_report_sql (void)`

Returns the SQL query to run the "all JEs" report.

Returns

The SQL query.

5.37 `lib/database/mysql/db_mysql_check_total_entity_report_sql.c` File Reference

Returns MYSQL SQL query to create check total report by entity.

Functions

- `const char * db_check_total_entity_report_sql (void)`
Returns the SQL query to run the "check total" by entity. report.

5.37.1 Detailed Description

Returns MYSQL SQL query to create check total report by entity.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.37.2 Function Documentation

5.37.2.1 `const char* db_check_total_entity_report_sql (void)`

Returns the SQL query to run the "check total" by entity. report.

Returns

The SQL query.

5.38 lib/database/mysql/db_mysql_check_total_report_sql.c File Reference

Returns MYSQL SQL query to create check total report.

Functions

- `const char * db_check_total_report_sql (void)`
Returns the SQL query to run the "check total" report.

5.38.1 Detailed Description

Returns MYSQL SQL query to create check total report.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.38.2 Function Documentation

5.38.2.1 `const char* db_check_total_report_sql (void)`

Returns the SQL query to run the "check total" report.

Returns

The SQL query.

5.39 lib/database/mysql/db_mysql_create_all_jes_view_sql.c File Reference

Returns MYSQL SQL query to create all_jes view.

Functions

- const char * [db_create_all_jes_view_sql](#) (void)
Returns the SQL query to create the all JEs view.

5.39.1 Detailed Description

Returns MYSQL SQL query to create all_jes view.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.39.2 Function Documentation

5.39.2.1 const char* db_create_all_jes_view_sql (void)

Returns the SQL query to create the all JEs view.

Returns

The SQL query.

5.40 lib/database/mysql/db_mysql_create_check_total_view_sql.c File Reference

Returns MYSQL SQL query to create check total view.

Functions

- const char * [db_create_check_total_view_sql](#) (void)
Returns the SQL query to create the check total view.

5.40.1 Detailed Description

Returns MYSQL SQL query to create check total view.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.40.2 Function Documentation

5.40.2.1 `const char* db_create_check_total_view_sql (void)`

Returns the SQL query to create the check total view.

Returns

The SQL query.

5.41 lib/database/mysql/db_mysql_create_current_trial_balance_view_sql.c File Reference

Returns MYSQL SQL query to create trial balance view.

Functions

- `const char * db_create_current_trial_balance_view_sql (void)`
Returns the SQL query to create the current TB view.

5.41.1 Detailed Description

Returns MYSQL SQL query to create trial balance view.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.41.2 Function Documentation

5.41.2.1 `const char* db_create_current_trial_balance_view_sql (void)`

Returns the SQL query to create the current TB view.

Returns

The SQL query.

5.42 lib/database/mysql/db_mysql_create_entities_table_sql.c File Reference

Returns MYSQL SQL query to create entities table.

Functions

- `const char * db_create_entities_table_sql (void)`
Returns the SQL query to create the entities table.

5.42.1 Detailed Description

Returns MYSQL SQL query to create entities table.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.42.2 Function Documentation

5.42.2.1 `const char* db_create_entities_table_sql (void)`

Returns the SQL query to create the entities table.

Returns

The SQL query.

5.43 lib/database/mysql/db_mysql_create_jelines_table_sql.c File Reference

Returns MYSQL SQL query to create journal entry lines table.

Functions

- `const char * db_create_jelines_table_sql (void)`
Returns the SQL query to create the JE lines table.

5.43.1 Detailed Description

Returns MYSQL SQL query to create journal entry lines table.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.43.2 Function Documentation

5.43.2.1 `const char* db_create_jelines_table_sql (void)`

Returns the SQL query to create the JE lines table.

Returns

The SQL query.

5.44 lib/database/mysql/db_mysql_create_jes_table_sql.c File Reference

Returns MYSQL SQL query to create journal entries table.

Functions

- const char * [db_create_jes_table_sql](#) (void)
Returns the SQL query to create the journal entries table.

5.44.1 Detailed Description

Returns MYSQL SQL query to create journal entries table.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.44.2 Function Documentation

5.44.2.1 const char* db_create_jes_table_sql (void)

Returns the SQL query to create the journal entries table.

Returns

The SQL query.

5.45 lib/database/mysql/db_mysql_create_jesrcs_table_sql.c File Reference

Returns MYSQL SQL query to create JE sources table.

Functions

- const char * [db_create_jesrcs_table_sql](#) (void)
Returns the SQL query to create the JE sources table.

5.45.1 Detailed Description

Returns MYSQL SQL query to create JE sources table.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.45.2 Function Documentation

5.45.2.1 `const char* db_create_jesrcs_table_sql (void)`

Returns the SQL query to create the JE sources table.

Returns

The SQL query.

5.46 `lib/database/mysql/db_mysql_create_nomaccts_table_sql.c` File Reference

Returns MYSQL SQL query to create nominal accounts table.

Functions

- `const char * db_create_nomaccts_table_sql (void)`
Returns the SQL query to create the nominal accounts table.

5.46.1 Detailed Description

Returns MYSQL SQL query to create nominal accounts table.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.46.2 Function Documentation

5.46.2.1 `const char* db_create_nomaccts_table_sql (void)`

Returns the SQL query to create the nominal accounts table.

Returns

The SQL query.

5.47 `lib/database/mysql/db_mysql_create_standingdata_table_sql.c` File Reference

Returns MYSQL SQL query to create standing data table.

Functions

- `const char * db_create_standingdata_table_sql (void)`
Returns the SQL query to create the standing data table.

5.47.1 Detailed Description

Returns MYSQL SQL query to create standing data table.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.47.2 Function Documentation

5.47.2.1 `const char* db_create_standingdata_table_sql (void)`

Returns the SQL query to create the standing data table.

Returns

The SQL query.

5.48 lib/database/mysql/db_mysql_create_users_table_sql.c File Reference

Returns MYSQL SQL query to create users table.

Functions

- `const char * db_create_users_table_sql (void)`
Returns the SQL query to create the users table.

5.48.1 Detailed Description

Returns MYSQL SQL query to create users table.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.48.2 Function Documentation

5.48.2.1 `const char* db_create_users_table_sql (void)`

Returns the SQL query to create the users table.

Returns

The SQL query.

5.49 lib/database/mysql/db_mysql_current_trial_balance_entity_report_sql.c File Reference

Returns MYSQL SQL query to create current TB by entity report.

Functions

- const char * [db_current_trial_balance_entity_report_sql](#) (void)
Returns the SQL query to run the "current TB" by entity. report.

5.49.1 Detailed Description

Returns MYSQL SQL query to create current TB by entity report.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.49.2 Function Documentation

5.49.2.1 const char* db_current_trial_balance_entity_report_sql (void)

Returns the SQL query to run the "current TB" by entity. report.

Returns

The SQL query.

5.50 lib/database/mysql/db_mysql_current_trial_balance_report_sql.c File Reference

Returns MYSQL SQL query to create current TB report.

Functions

- const char * [db_current_trial_balance_report_sql](#) (void)
Returns the SQL query to run the "current TB" report.

5.50.1 Detailed Description

Returns MYSQL SQL query to create current TB report.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.50.2 Function Documentation

5.50.2.1 `const char* db_current_trial_balance_report_sql (void)`

Returns the SQL query to run the "current TB" report.

Returns

The SQL query.

5.51 lib/database/mysql/db_mysql_drop_all_jes_view_sql.c File Reference

Returns MYSQL SQL query to drop all JES view.

Functions

- `const char * db_drop_all_jes_view_sql (void)`
Returns the SQL query to drop the all JEs view.

5.51.1 Detailed Description

Returns MYSQL SQL query to drop all JES view.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.51.2 Function Documentation

5.51.2.1 `const char* db_drop_all_jes_view_sql (void)`

Returns the SQL query to drop the all JEs view.

Returns

The SQL query.

5.52 lib/database/mysql/db_mysql_drop_check_total_view_sql.c File Reference

Returns MYSQL SQL query to drop check total view.

Functions

- `const char * db_drop_check_total_view_sql (void)`
Returns the SQL query to drop the check total view.

5.52.1 Detailed Description

Returns MYSQL SQL query to drop check total view.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.52.2 Function Documentation

5.52.2.1 `const char* db_drop_check_total_view_sql (void)`

Returns the SQL query to drop the check total view.

Returns

The SQL query.

5.53 `lib/database/mysql/db_mysql_drop_current_trial_balance_view_sql.c` File Reference

Returns MYSQL SQL query to drop trial balance view.

Functions

- `const char * db_drop_current_trial_balance_view_sql (void)`
Returns the SQL query to drop the current TB view.

5.53.1 Detailed Description

Returns MYSQL SQL query to drop trial balance view.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.53.2 Function Documentation

5.53.2.1 `const char* db_drop_current_trial_balance_view_sql (void)`

Returns the SQL query to drop the current TB view.

Returns

The SQL query.

5.54 lib/database/mysql/db_mysql_drop_entities_table_sql.c File Reference

Returns MYSQL SQL query to drop entities table.

Functions

- const char * [db_drop_entities_table_sql](#) (void)
Returns the SQL query to drop the entities table.

5.54.1 Detailed Description

Returns MYSQL SQL query to drop entities table.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.54.2 Function Documentation

5.54.2.1 const char* db_drop_entities_table_sql (void)

Returns the SQL query to drop the entities table.

Returns

The SQL query.

5.55 lib/database/mysql/db_mysql_drop_jelines_table_sql.c File Reference

Returns MYSQL SQL query to drop journal entry lines table.

Functions

- const char * [db_drop_jelines_table_sql](#) (void)
Returns the SQL query to drop the JE lines table.

5.55.1 Detailed Description

Returns MYSQL SQL query to drop journal entry lines table.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.55.2 Function Documentation

5.55.2.1 `const char* db_drop_jelines_table_sql (void)`

Returns the SQL query to drop the JE lines table.

Returns

The SQL query.

5.56 `lib/database/mysql/db_mysql_drop_jes_table_sql.c` File Reference

Returns MYSQL SQL query to drop entities table.

Functions

- `const char * db_drop_jes_table_sql (void)`
Returns the SQL query to drop the journal entries table.

5.56.1 Detailed Description

Returns MYSQL SQL query to drop entities table.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.56.2 Function Documentation

5.56.2.1 `const char* db_drop_jes_table_sql (void)`

Returns the SQL query to drop the journal entries table.

Returns

The SQL query.

5.57 `lib/database/mysql/db_mysql_drop_jesrcs_table_sql.c` File Reference

Returns MYSQL SQL query to drop JE sources table.

Functions

- `const char * db_drop_jesrcs_table_sql (void)`
Returns the SQL query to drop the JE sources table.

5.57.1 Detailed Description

Returns MYSQL SQL query to drop JE sources table.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.57.2 Function Documentation

5.57.2.1 `const char* db_drop_jesrcs_table_sql (void)`

Returns the SQL query to drop the JE sources table.

Returns

The SQL query.

5.58 lib/database/mysql/db_mysql_drop_nomaccts_table_sql.c File Reference

Returns MYSQL SQL query to drop nominal accounts table.

Functions

- `const char * db_drop_nomaccts_table_sql (void)`
Returns the SQL query to drop the nominal accounts table.

5.58.1 Detailed Description

Returns MYSQL SQL query to drop nominal accounts table.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.58.2 Function Documentation

5.58.2.1 `const char* db_drop_nomaccts_table_sql (void)`

Returns the SQL query to drop the nominal accounts table.

Returns

The SQL query.

5.59 lib/database/mysql/db_mysql_drop_standingdata_table_sql.c File Reference

Returns MYSQL SQL query to drop standing data table.

Functions

- `const char * db_drop_standingdata_table_sql (void)`
Returns the SQL query to drop the standing data table.

5.59.1 Detailed Description

Returns MYSQL SQL query to drop standing data table.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.59.2 Function Documentation

5.59.2.1 `const char* db_drop_standingdata_table_sql (void)`

Returns the SQL query to drop the standing data table.

Returns

The SQL query.

5.60 lib/database/mysql/db_mysql_drop_users_table_sql.c File Reference

Returns MYSQL SQL query to drop users table.

Functions

- `const char * db_drop_users_table_sql (void)`
Returns the SQL query to drop the users table.

5.60.1 Detailed Description

Returns MYSQL SQL query to drop users table.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.60.2 Function Documentation

5.60.2.1 `const char* db_drop_users_table_sql (void)`

Returns the SQL query to drop the users table.

Returns

The SQL query.

5.61 lib/database/mysql/db_mysql_entity_name_from_id_sql.c File Reference

Returns MYSQL SQL query to fetch an entity name from its ID.

Functions

- `const char * db_get_entity_name_from_id_sql (void)`
Returns the SQL query to get an entity name from its ID.

5.61.1 Detailed Description

Returns MYSQL SQL query to fetch an entity name from its ID.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.61.2 Function Documentation

5.61.2.1 `const char* db_get_entity_name_from_id_sql (void)`

Returns the SQL query to get an entity name from its ID.

\

Returns

The SQL query.

5.62 lib/database/mysql/db_mysql_general.c File Reference

Implementation of MYSQL database functionality.

```
#include <my_global.h>
#include <my_sys.h>
#include <mysql.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "gl_general/gl_general.h"
#include "database/db_internal.h"
```


Returns

`true` if the connection was successfully made, `false` otherwise.

5.62.2.2 ds_recordset db_create_recordset_from_query (ds_str query)

Creates a [ds_recordset](#) from a query.

Parameters

<i>query</i>	The SELECT query to run.
--------------	--------------------------

Returns

A [ds_recordset](#) containing the query result, or `NULL` on failure.

5.62.2.3 bool db_execute_query (ds_str query)

Executes an SQL query on the database.

Parameters

<i>query</i>	The query to execute.
--------------	-----------------------

Returns

`true` if the query was successfully executed, `false` otherwise.

5.62.3 Variable Documentation**5.62.3.1 MYSQL* conn_mss = NULL**

MYSQL connection object.

5.62.3.2 MYSQL* main_mss = NULL

MYSQL initialization object.

5.63 lib/database/mysql/db_mysql_list_entities_report_sql.c File Reference

Returns MYSQL SQL query to create list entities report.

Functions

- `const char * db_list_entities_report_sql (void)`
Returns the SQL query to run the "list entities" report.

5.63.1 Detailed Description

Returns MYSQL SQL query to create list entities report.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.63.2 Function Documentation**5.63.2.1 `const char* db_list_entities_report_sql (void)`**

Returns the SQL query to run the "list entities" report.

Returns

The SQL query.

5.64 `lib/database/mysql/db_mysql_list_jelines_report_sql.c` File Reference

Returns MYSQL SQL query to create JE lines report.

Functions

- `const char * db_list_jelines_report_sql (void)`
Returns the SQL query to run the "list JE lines" report.

5.64.1 Detailed Description

Returns MYSQL SQL query to create JE lines report.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.64.2 Function Documentation**5.64.2.1 `const char* db_list_jelines_report_sql (void)`**

Returns the SQL query to run the "list JE lines" report.

Returns

The SQL query.

5.65 `lib/database/mysql/db_mysql_list_jes_report_sql.c` File Reference

Returns MYSQL SQL query to create journal entries report.

Functions

- const char * [db_list_jes_report_sql](#) (void)
Returns the SQL query to run the "list journal entries" report.

5.65.1 Detailed Description

Returns MYSQL SQL query to create journal entries report.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.65.2 Function Documentation

5.65.2.1 const char* db_list_jes_report_sql (void)

Returns the SQL query to run the "list journal entries" report.

Returns

The SQL query.

5.66 lib/database/mysql/db_mysql_list_jesrcs_report_sql.c File Reference

Returns MYSQL SQL query to create JE sources report.

Functions

- const char * [db_list_jesrcs_report_sql](#) (void)
Returns the SQL query to run the "list JE sources" report.

5.66.1 Detailed Description

Returns MYSQL SQL query to create JE sources report.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.66.2 Function Documentation

5.66.2.1 `const char* db_list_jesrcs_report_sql (void)`

Returns the SQL query to run the "list JE sources" report.

Returns

The SQL query.

5.67 `lib/database/mysql/db_mysql_list_nomaccts_report_sql.c` File Reference

Returns MYSQL SQL query to create list nominal accounts report.

Functions

- `const char * db_list_nomaccts_report_sql (void)`
Returns the SQL query to run the "list nominal accounts" report.

5.67.1 Detailed Description

Returns MYSQL SQL query to create list nominal accounts report.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.67.2 Function Documentation

5.67.2.1 `const char* db_list_nomaccts_report_sql (void)`

Returns the SQL query to run the "list nominal accounts" report.

Returns

The SQL query.

5.68 `lib/database/mysql/db_mysql_list_users_report_sql.c` File Reference

Returns MYSQL SQL query to create list users report.

Functions

- `const char * db_list_users_report_sql (void)`
Returns the SQL query to run the "list users" report.

5.68.1 Detailed Description

Returns MYSQL SQL query to create list users report.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.68.2 Function Documentation

5.68.2.1 `const char* db_list_users_report_sql (void)`

Returns the SQL query to run the "list users" report.

Returns

The SQL query.

5.69 lib/database/mysql/db_mysql_show_standingdata_report_sql.c File Reference

Returns MYSQL SQL query to create show standing data report.

Functions

- `const char * db_show_standingdata_report_sql (void)`
Returns the SQL query to run the "show standing data" report.

5.69.1 Detailed Description

Returns MYSQL SQL query to create show standing data report.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.69.2 Function Documentation

5.69.2.1 `const char* db_show_standingdata_report_sql (void)`

Returns the SQL query to run the "show standing data" report.

Returns

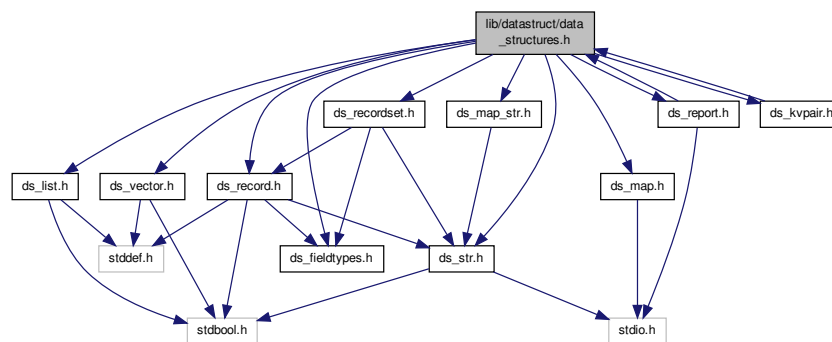
The SQL query.

5.70 lib/datastruct/data_structures.h File Reference

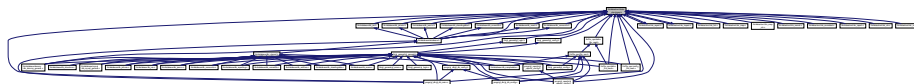
Interface to data structures.

```
#include "ds_list.h"
#include "ds_vector.h"
#include "ds_str.h"
#include "ds_map.h"
#include "ds_map_str.h"
#include "ds_fieldtypes.h"
#include "ds_record.h"
#include "ds_recordset.h"
#include "ds_report.h"
#include "ds_kvpair.h"
```

Include dependency graph for data_structures.h:



This graph shows which files directly or indirectly include this file:



5.70.1 Detailed Description

Interface to data structures.

Author

Paul Griffiths

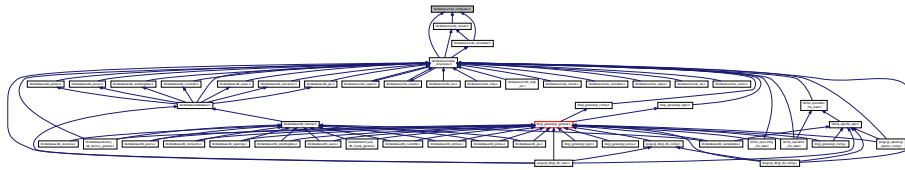
Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.71 lib/datastruct/ds_fieldtypes.h File Reference

Record field types enumeration.

This graph shows which files directly or indirectly include this file:



Enumerations

- enum `ds_field_types` { `DS_FIELD_STRING`, `DS_FIELD_INT`, `DS_FIELD_BOOLEAN`, `DS_FIELD_DOUBLE` }

5.71.1 Detailed Description

Record field types enumeration.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.71.2 Enumeration Type Documentation

5.71.2.1 enum `ds_field_types`

Enumeration for field type

Enumerator:

`DS_FIELD_STRING` Field is string type

`DS_FIELD_INT` Field is integer type

`DS_FIELD_BOOLEAN` Field is boolean type

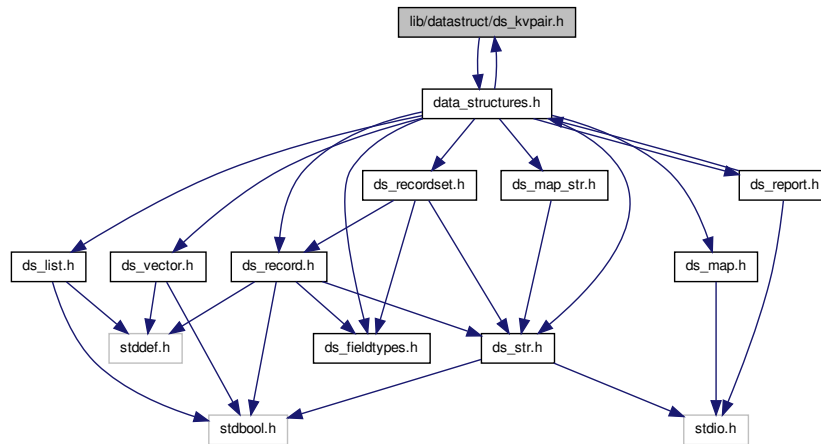
`DS_FIELD_DOUBLE` Field is double type

5.72 lib/datastruct/ds_kvpair.h File Reference

Implementation of string key-value pair data structure.

```
#include "data_structures.h"
```

Include dependency graph for ds_kvpair.h:



This graph shows which files directly or indirectly include this file:



Typedefs

- typedef struct [ds_kvpair](#) * [ds_kvpair](#)

Functions

- [ds_kvpair](#) [ds_kvpair_create](#) ([ds_str](#) key, [ds_str](#) value)
Makes a new key-value pair node.
- void [ds_kvpair_destroy](#) ([ds_kvpair](#) pair)
Frees a key-value pair.
- void [ds_kvpair_destructor](#) (void *pair)
Destructor function.
- [ds_str](#) [ds_kvpair_get_key](#) ([ds_kvpair](#) pair)
Returns the key in a key-value pair.
- [ds_str](#) [ds_kvpair_get_value](#) ([ds_kvpair](#) pair)
Returns the value in a key-value pair.

5.72.1 Detailed Description

Implementation of string key-value pair data structure. Interface to string key-value pair data structure.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

Author

Paul Griffiths

Copyright

Copyright 2013 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.72.2 Typedef Documentation

5.72.2.1 typedef struct ds_kvpair* ds_kvpair

Opaque data type for kvpair

5.72.3 Function Documentation

5.72.3.1 ds_kvpair ds_kvpair_create (ds_str key, ds_str value)

Makes a new key-value pair node.

Parameters

<i>key</i>	The key.
<i>value</i>	The value.

Returns

A pointer to the newly-created node.

5.72.3.2 void ds_kvpair_destroy (ds_kvpair pair)

Frees a key-value pair.

Parameters

<i>pair</i>	The key-value pair to free.
-------------	-----------------------------

5.72.3.3 void ds_kvpair_destructor (void * pair)

Destructor function.

This function is useful for passing a destructor function to a data structure which expects a prototype conforming to `free()`.

Parameters

<i>pair</i>	A pointer to the node to free.
-------------	--------------------------------

5.72.3.4 `ds_str ds_kvpair_get_key (ds_kvpair pair)`

Returns the key in a key-value pair.

Parameters

<i>pair</i>	The key-value pair.
-------------	---------------------

Returns

The key. The caller should not destroy this string.

5.72.3.5 `ds_str ds_kvpair_get_value (ds_kvpair pair)`

Returns the value in a key-value pair.

Parameters

<i>pair</i>	The key-value pair.
-------------	---------------------

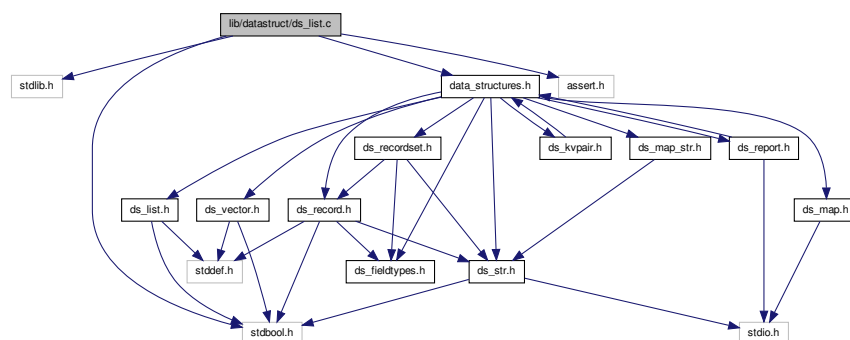
Returns

The value. The caller should not destroy this string.

5.73 `lib/datastruct/ds_list.c` File Reference

Implementation of generic doubly-linked list data structure.

```
#include <stdlib.h>
#include <stdbool.h>
#include <assert.h>
#include "data_structures.h"
Include dependency graph for ds_list.c:
```



Data Structures

- struct [ds_list_element](#)
- struct [ds_list](#)

Functions

- `ds_list ds_list_create` (const bool free_on_delete, void(*destructor)(void *))
Creates a new list.
- void `ds_list_destroy` (ds_list list)
Destroys a list and frees any associated resources.
- void `ds_list_destructor` (void *list)
A list destructor function.
- `ds_list ds_list_append` (ds_list list, void *data)
Appends an element to a list.
- void `ds_list_remove_tail` (ds_list list)
Removes the last element of a list.
- void `ds_list_remove_all` (ds_list list)
Removes all the elements from a list.
- void * `ds_list_element` (ds_list list, const size_t index)
Retrieves the data at a specified index.
- size_t `ds_list_length` (ds_list list)
Returns the number of elements in a list.
- bool `ds_list_is_empty` (ds_list list)
Checks if a list is empty.
- void `ds_list_seek_start` (ds_list list)
Sets the current element to the first element of a list.
- void `ds_list_seek_end` (ds_list list)
Sets the current element to the last element of a list.
- void * `ds_list_get_next_data` (ds_list list)
Returns the next element of the list.
- void * `ds_list_get_prev_data` (ds_list list)
Returns the previous element of the list.

5.73.1 Detailed Description

Implementation of generic doubly-linked list data structure.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.73.2 Function Documentation

5.73.2.1 ds_list ds_list_append (ds_list list, void * element)

Appends an element to a list.

Parameters

<i>list</i>	The list to which to append.
<i>element</i>	The element to append.

Returns

The same list, or `NULL` on failure.

5.73.2.2 `ds_list ds_list_create (const bool free_on_delete, void(*)(void *) destructor)`

Creates a new list.

Parameters

<i>free_on_delete</i>	Set to <code>true</code> if the list elements should be destroyed when removed from the list, and when the list itself is destroyed. If set to <code>false</code> , the caller is responsible for destroying the elements prior to destroying the list.
<i>destructor</i>	Pointer to a destructor function to use for destroying the list elements, when <code>free_on_delete</code> is true. If this is set to <code>NULL</code> , <code>free()</code> from the standard C library will be used to destroy the elements.

Returns

A newly created list, or `NULL` on failure.

5.73.2.3 `void ds_list_destroy (ds_list list)`

Destroys a list and frees any associated resources.

Parameters

<i>list</i>	The list to destroy.
-------------	----------------------

5.73.2.4 `void ds_list_destructor (void * list)`

A list destructor function.

This function may be passed to `ds_list_create()` when creating a list of lists. It calls `ds_list_destroy()`, but the parameter of `ds_list_destroy()` is not compatible with the function signature expected by `ds_list_create()`, so this function provides an appropriate interface.

Parameters

<i>list</i>	The list to destroy.
-------------	----------------------

5.73.2.5 `void* ds_list_element (ds_list list, const size_t index)`

Retrieves the data at a specified index.

Parameters

<i>list</i>	The list from which to retrieve.
<i>index</i>	The index of the desired element.

Returns

A pointer to the data, or `NULL` if the index is out of range.

5.73.2.6 void* ds_list_get_next_data (ds_list list)

Returns the next element of the list.

This function returns the data of the "current element", and advances the current element pointer. Subsequent calls to this function will return successive elements.

Parameters

<i>list</i>	The list.
-------------	-----------

Returns

A pointer to the next element, or `NULL` if the end of the list has been reached.

5.73.2.7 void* ds_list_get_prev_data (ds_list list)

Returns the previous element of the list.

This function returns the data of the "current element", and decrements the current element pointer. Subsequent calls to this function will return successively earlier elements.

Parameters

<i>list</i>	The list.
-------------	-----------

Returns

A pointer to the previous element, or `NULL` if the start of the list has been reached.

5.73.2.8 bool ds_list_is_empty (ds_list list)

Checks if a list is empty.

Parameters

<i>list</i>	The list to check.
-------------	--------------------

Returns

`true` if the list is empty, `false` otherwise.

5.73.2.9 size_t ds_list_length (ds_list list)

Returns the number of elements in a list.

Parameters

<i>list</i>	The list.
-------------	-----------

Returns

The number of elements in the list.

5.73.2.10 void ds_list_remove_all (ds_list list)

Removes all the elements from a list.

Parameters

<i>list</i>	The list from which to remove.
-------------	--------------------------------

5.73.2.11 void ds_list_remove_tail (ds_list list)

Removes the last element of a list.

Parameters

<i>list</i>	The list from which to remove.
-------------	--------------------------------

5.73.2.12 void ds_list_seek_end (ds_list list)

Sets the current element to the last element of a list.

Parameters

<i>list</i>	The list.
-------------	-----------

5.73.2.13 void ds_list_seek_start (ds_list list)

Sets the current element to the first element of a list.

Parameters

<i>list</i>	The list.
-------------	-----------

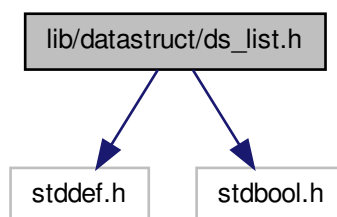
5.74 lib/datastruct/ds_list.h File Reference

Interface to generic doubly-linked list data structure.

```
#include <stddef.h>
```

```
#include <stdbool.h>
```

Include dependency graph for ds_list.h:



This graph shows which files directly or indirectly include this file:



Typedefs

- typedef struct [ds_list](#) * [ds_list](#)

Functions

- [ds_list ds_list_create](#) (const bool free_on_delete, void(*destructor)(void *))
Creates a new list.
- void [ds_list_destroy](#) ([ds_list](#) list)
Destroys a list and frees any associated resources.
- void [ds_list_destructor](#) (void *list)
A list destructor function.
- [ds_list ds_list_append](#) ([ds_list](#) list, void *element)
Appends an element to a list.
- void [ds_list_remove_tail](#) ([ds_list](#) list)
Removes the last element of a list.
- void [ds_list_remove_all](#) ([ds_list](#) list)
Removes all the elements from a list.
- void * [ds_list_element](#) ([ds_list](#) list, const size_t index)
Retrieves the data at a specified index.
- size_t [ds_list_length](#) ([ds_list](#) list)
Returns the number of elements in a list.
- bool [ds_list_is_empty](#) ([ds_list](#) list)
Checks if a list is empty.
- void [ds_list_seek_start](#) ([ds_list](#) list)
Sets the current element to the first element of a list.
- void [ds_list_seek_end](#) ([ds_list](#) list)
Sets the current element to the last element of a list.
- void * [ds_list_get_next_data](#) ([ds_list](#) list)
Returns the next element of the list.
- void * [ds_list_get_prev_data](#) ([ds_list](#) list)
Returns the previous element of the list.

5.74.1 Detailed Description

Interface to generic doubly-linked list data structure.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.74.2 Typedef Documentation

5.74.2.1 typedef struct ds_list* ds_list

Typedef for opaque list datatype

5.74.3 Function Documentation

5.74.3.1 ds_list ds_list_append (ds_list *list*, void * *element*)

Appends an element to a list.

Parameters

<i>list</i>	The list to which to append.
<i>element</i>	The element to append.

Returns

The same list, or `NULL` on failure.

5.74.3.2 ds_list ds_list_create (const bool *free_on_delete*, void(*)(void *) *destructor*)

Creates a new list.

Parameters

<i>free_on_delete</i>	Set to <code>true</code> if the list elements should be destroyed when removed from the list, and when the list itself is destroyed. If set to <code>false</code> , the caller is responsible for destroying the elements prior to destroying the list.
<i>destructor</i>	Pointer to a destructor function to use for destroying the list elements, when <code>free_on_delete</code> is true. If this is set to <code>NULL</code> , <code>free()</code> from the standard C library will be used to destroy the elements.

Returns

A newly created list, or `NULL` on failure.

5.74.3.3 void ds_list_destroy (ds_list *list*)

Destroys a list and frees any associated resources.

Parameters

<i>list</i>	The list to destroy.
-------------	----------------------

5.74.3.4 void ds_list_destructor (void * *list*)

A list destructor function.

This function may be passed to `ds_list_create()` when creating a list of lists. It calls `ds_list_destroy()`, but the parameter of `ds_list_destroy()` is not compatible with the function signature expected by `ds_list_create()`, so this function provides an appropriate interface.

Parameters

<i>list</i>	The list to destroy.
-------------	----------------------

5.74.3.5 void* ds_list_element (ds_list list, const size_t index)

Retrieves the data at a specified index.

Parameters

<i>list</i>	The list from which to retrieve.
<i>index</i>	The index of the desired element.

Returns

A pointer to the data, or `NULL` if the index is out of range.

5.74.3.6 void* ds_list_get_next_data (ds_list list)

Returns the next element of the list.

This function returns the data of the "current element", and advances the current element pointer. Subsequent calls to this function will return successive elements.

Parameters

<i>list</i>	The list.
-------------	-----------

Returns

A pointer to the next element, or `NULL` if the end of the list has been reached.

5.74.3.7 void* ds_list_get_prev_data (ds_list list)

Returns the previous element of the list.

This function returns the data of the "current element", and decrements the current element pointer. Subsequent calls to this function will return successively earlier elements.

Parameters

<i>list</i>	The list.
-------------	-----------

Returns

A pointer to the previous element, or `NULL` if the start of the list has been reached.

5.74.3.8 bool ds_list_is_empty (ds_list list)

Checks if a list is empty.

Parameters

<i>list</i>	The list to check.
-------------	--------------------

Returns

`true` if the list is empty, `false` otherwise.

5.74.3.9 `size_t ds_list_length (ds_list list)`

Returns the number of elements in a list.

Parameters

<i>list</i>	The list.
-------------	-----------

Returns

The number of elements in the list.

5.74.3.10 `void ds_list_remove_all (ds_list list)`

Removes all the elements from a list.

Parameters

<i>list</i>	The list from which to remove.
-------------	--------------------------------

5.74.3.11 `void ds_list_remove_tail (ds_list list)`

Removes the last element of a list.

Parameters

<i>list</i>	The list from which to remove.
-------------	--------------------------------

5.74.3.12 `void ds_list_seek_end (ds_list list)`

Sets the current element to the last element of a list.

Parameters

<i>list</i>	The list.
-------------	-----------

5.74.3.13 `void ds_list_seek_start (ds_list list)`

Sets the current element to the first element of a list.

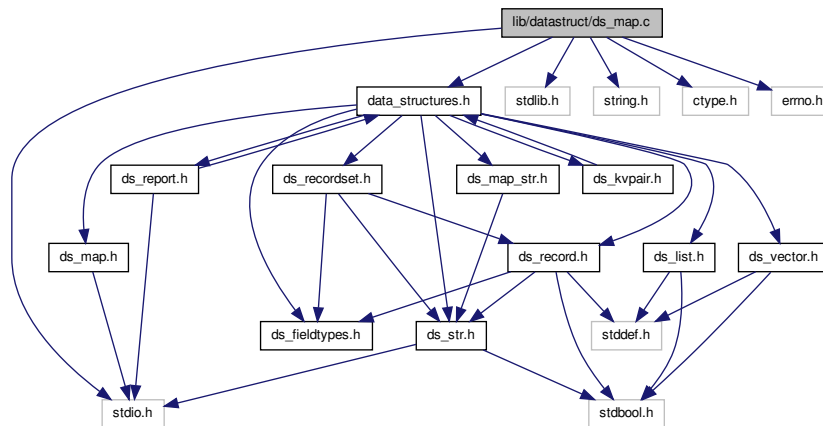
Parameters

<i>list</i>	The list.
-------------	-----------

5.75 `lib/datastruct/ds_map.c` File Reference

Implementation of string-string hash map data structure.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#include <errno.h>
#include "data_structures.h"
Include dependency graph for ds_map.c:
```



Data Structures

- struct [kv_pair_node](#)
- struct [ds_map](#)

Macros

- `#define _POSIX_C_SOURCE 200809L`
Enables POSIX library functions.

Functions

- `ds_map ds_map_init (const size_t hash_size)`
Initializes a hash map.
- `void ds_map_destroy (ds_map map)`
Destroys a hash map.
- `const char * ds_map_get_value (ds_map map, const char *key)`
Retrieves a value associated with a key in the map.
- `void ds_map_insert (ds_map map, const char *key, const char *value)`
Inserts a key-value pair into a map.
- `void ds_map_print_all (ds_map map, FILE *outfile)`
Prints all the key-value pairs in a map to stdout.

5.75.1 Detailed Description

Implementation of string-string hash map data structure.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.75.2 Function Documentation**5.75.2.1 void ds_map_destroy (ds_map map)**

Destroys a hash map.

Parameters

<i>map</i>	A reference to the map to destroy.
------------	------------------------------------

5.75.2.2 const char* ds_map_get_value (ds_map map, const char * key)

Retrieves a value associated with a key in the map.

Parameters

<i>map</i>	A reference to the hash map.
<i>key</i>	The key.

Returns

A pointer to the value associated with the key, or `NULL` if the key is not in the map. The caller should not modify the string to which this pointer points.

5.75.2.3 ds_map ds_map_init (const size_t hash_size)

Initializes a hash map.

Parameters

<i>hash_size</i>	The number of possible hash values.
------------------	-------------------------------------

Returns

A reference to the newly-created hash map.

5.75.2.4 void ds_map_insert (ds_map map, const char * key, const char * value)

Inserts a key-value pair into a map.

The key and value are copied, so the caller may modify or `free()` them after calling this function.

Parameters

<i>map</i>	A reference to the hash map.
<i>key</i>	The key.
<i>value</i>	The value.

5.75.2.5 void ds_map_print_all (ds_map map, FILE * outfile)

Prints all the key-value pairs in a map to stdout.

Parameters

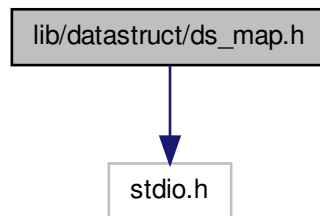
<i>map</i>	A reference to the map.
<i>outfile</i>	A FILE pointer to which to print the output.

5.76 lib/datastruct/ds_map.h File Reference

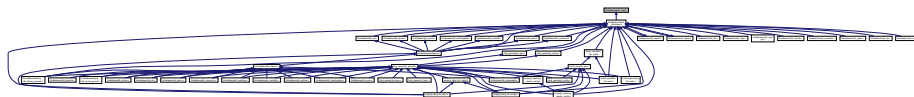
Interface to string-string hash map data structure.

```
#include <stdio.h>
```

Include dependency graph for ds_map.h:



This graph shows which files directly or indirectly include this file:



Typedefs

- typedef struct [ds_map](#) * [ds_map](#)

Functions

- [ds_map ds_map_init](#) (const size_t hash_size)
Initializes a hash map.
- void [ds_map_destroy](#) ([ds_map](#) map)
Destroys a hash map.
- const char * [ds_map_get_value](#) ([ds_map](#) map, const char *key)
Retrieves a value associated with a key in the map.
- void [ds_map_insert](#) ([ds_map](#) map, const char *key, const char *value)
Inserts a key-value pair into a map.
- void [ds_map_print_all](#) ([ds_map](#) map, FILE *outfile)
Prints all the key-value pairs in a map to stdout.

5.76.1 Detailed Description

Interface to string-string hash map data structure.

Author

Paul Griffiths

Copyright

Copyright 2013 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.76.2 Typedef Documentation

5.76.2.1 typedef struct **ds_map*** **ds_map**

Opaque data type for hash map

5.76.3 Function Documentation

5.76.3.1 void **ds_map_destroy** (**ds_map** *map*)

Destroys a hash map.

Parameters

<i>map</i>	A reference to the map to destroy.
------------	------------------------------------

5.76.3.2 const char* **ds_map_get_value** (**ds_map** *map*, const char * *key*)

Retrieves a value associated with a key in the map.

Parameters

<i>map</i>	A reference to the hash map.
<i>key</i>	The key.

Returns

A pointer to the value associated with the key, or `NULL` if the key is not in the map. The caller should not modify the string to which this pointer points.

5.76.3.3 **ds_map** **ds_map_init** (const size_t *hash_size*)

Initializes a hash map.

Parameters

<i>hash_size</i>	The number of possible hash values.
------------------	-------------------------------------

Returns

A reference to the newly-created hash map.

5.76.3.4 void ds_map_insert (ds_map map, const char * key, const char * value)

Inserts a key-value pair into a map.

The key and value are copied, so the caller may modify or `free()` them after calling this function.

Parameters

<i>map</i>	A reference to the hash map.
<i>key</i>	The key.
<i>value</i>	The value.

5.76.3.5 void ds_map_print_all (ds_map map, FILE * outfile)

Prints all the key-value pairs in a map to stdout.

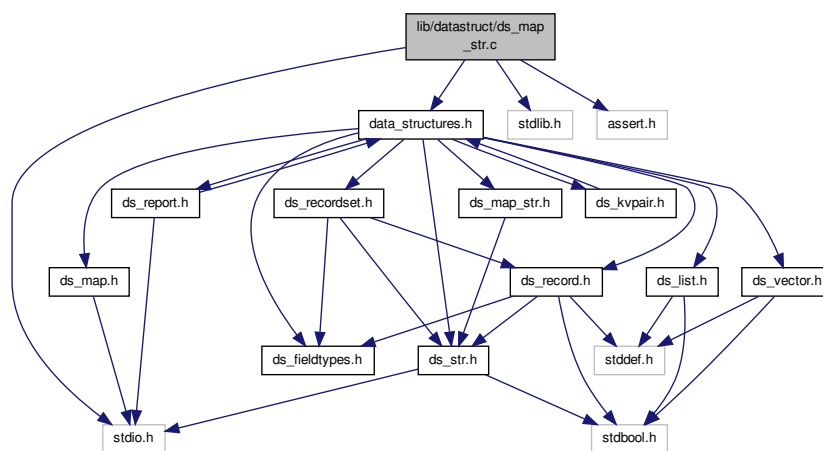
Parameters

<i>map</i>	A reference to the map.
<i>outfile</i>	A FILE pointer to which to print the output.

5.77 lib/datastruct/ds_map_str.c File Reference

Implementation of string-string hash map data structure.

```
#include <stdio.h>
#include <stdlib.h>
#include <assert.h>
#include "data_structures.h"
Include dependency graph for ds_map_str.c:
```



Data Structures

- struct [ds_map_str](#)

Functions

- [ds_map_str ds_map_str_init](#) (const size_t hash_size)
Initializes a hash map.
- void [ds_map_str_destroy](#) ([ds_map_str](#) map)
Destroys a hash map.
- [ds_str ds_map_str_get_value](#) ([ds_map_str](#) map, [ds_str](#) key)
Retrieves a value associated with a key in the map.
- void [ds_map_str_insert](#) ([ds_map_str](#) map, [ds_str](#) key, [ds_str](#) value)
Inserts a key-value pair into a map.

5.77.1 Detailed Description

Implementation of string-string hash map data structure.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.77.2 Function Documentation

5.77.2.1 void ds_map_str_destroy (ds_map_str map)

Destroys a hash map.

Parameters

<i>map</i>	A reference to the map to destroy.
------------	------------------------------------

5.77.2.2 ds_str ds_map_str_get_value (ds_map_str map, ds_str key)

Retrieves a value associated with a key in the map.

Parameters

<i>map</i>	A reference to the hash map.
<i>key</i>	The key.

Returns

A pointer to the value associated with the key, or `NULL` if the key is not in the map. The caller should not modify the string to which this pointer points.

5.77.2.3 ds_map_str ds_map_str_init (const size_t hash_size)

Initializes a hash map.

Parameters

<i>hash_size</i>	The number of possible hash values.
------------------	-------------------------------------

Returns

A reference to the newly-created hash map.

5.77.2.4 void ds_map_str_insert (ds_map_str map, ds_str key, ds_str value)

Inserts a key-value pair into a map.

The key and value are copied, so the caller may modify or `free()` them after calling this function.

Parameters

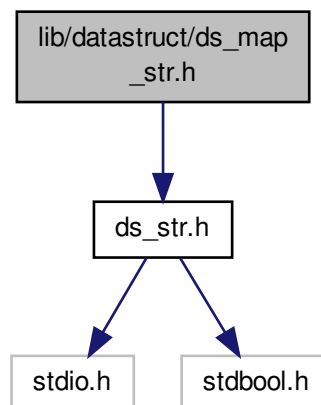
<i>map</i>	A reference to the hash map.
<i>key</i>	The key.
<i>value</i>	The value.

5.78 lib/datastruct/ds_map_str.h File Reference

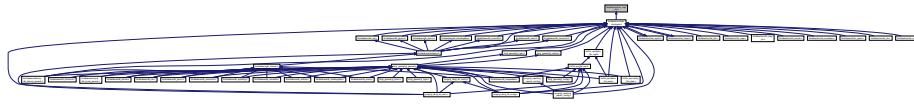
Interface to string-string hash map data structure.

```
#include "ds_str.h"
```

Include dependency graph for ds_map_str.h:



This graph shows which files directly or indirectly include this file:



Typedefs

- typedef struct [ds_map_str](#) * [ds_map_str](#)

Functions

- [ds_map_str ds_map_str_init](#) (const size_t hash_size)
Initializes a hash map.
- void [ds_map_str_destroy](#) ([ds_map_str](#) map)
Destroys a hash map.
- [ds_str ds_map_str_get_value](#) ([ds_map_str](#) map, [ds_str](#) key)
Retrieves a value associated with a key in the map.
- void [ds_map_str_insert](#) ([ds_map_str](#) map, [ds_str](#) key, [ds_str](#) value)
Inserts a key-value pair into a map.

5.78.1 Detailed Description

Interface to string-string hash map data structure.

Author

Paul Griffiths

Copyright

Copyright 2013 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.78.2 Typedef Documentation

5.78.2.1 typedef struct [ds_map_str](#)* [ds_map_str](#)

Opaque data type for hash map

5.78.3 Function Documentation

5.78.3.1 void [ds_map_str_destroy](#) ([ds_map_str](#) map)

Destroys a hash map.

Parameters

<i>map</i>	A reference to the map to destroy.
------------	------------------------------------

5.78.3.2 `ds_str ds_map_str_get_value (ds_map_str map, ds_str key)`

Retrieves a value associated with a key in the map.

Parameters

<i>map</i>	A reference to the hash map.
<i>key</i>	The key.

Returns

A pointer to the value associated with the key, or `NULL` if the key is not in the map. The caller should not modify the string to which this pointer points.

5.78.3.3 `ds_map_str ds_map_str_init (const size_t hash_size)`

Initializes a hash map.

Parameters

<i>hash_size</i>	The number of possible hash values.
------------------	-------------------------------------

Returns

A reference to the newly-created hash map.

5.78.3.4 `void ds_map_str_insert (ds_map_str map, ds_str key, ds_str value)`

Inserts a key-value pair into a map.

The key and value are copied, so the caller may modify or `free()` them after calling this function.

Parameters

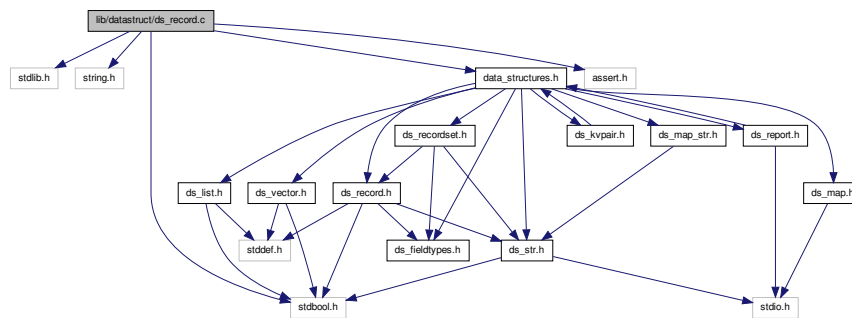
<i>map</i>	A reference to the hash map.
<i>key</i>	The key.
<i>value</i>	The value.

5.79 lib/datastruct/ds_record.c File Reference

Implementation of record database structure.

```
#include <stdlib.h>
#include <string.h>
#include <stdbool.h>
#include <assert.h>
#include "data_structures.h"
```

Include dependency graph for `ds_record.c`:



Data Structures

- struct `ds_record`

Functions

- `ds_record ds_record_create` (const size_t size)
Creates a new record.
- void `ds_record_destroy` (ds_record record)
Destroys a record and frees any associated resources.
- void `ds_record_destructor` (void *record)
A record destructor function.
- void `ds_record_clear` (ds_record record)
Clears and `free()`s all the elements in a record.
- void `ds_record_set_field` (ds_record record, const size_t index, ds_str field)
Sets a field of a record.
- ds_str `ds_record_get_field` (ds_record record, const size_t index)
Retrieves the field at a specified index.
- size_t `ds_record_size` (ds_record record)
Returns the size of a record.
- void `ds_record_seek_start` (ds_record record)
Sets the current field to the first field of a record.
- ds_str `ds_record_get_next_data` (ds_record record)
Returns the next field of the record.
- ds_record `ds_record_tokenize` (ds_str str, const char delim)
Tokenizes a string into a record.
- ds_str `ds_record_make_delim_string` (ds_record record, const char delim)
Makes a delimited string from a record.
- ds_str `ds_record_make_values_string` (ds_record record, enum ds_field_types *types)
Makes a delimited SQL values string from a record.

5.79.1 Detailed Description

Implementation of record database structure.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.79.2 Function Documentation

5.79.2.1 void ds_record_clear (ds_record record)

Clears and `free()`s all the elements in a record.

Parameters

<i>record</i>	The record.
---------------	-------------

5.79.2.2 ds_record ds_record_create (const size_t size)

Creates a new record.

Parameters

<i>size</i>	The size of the record.
-------------	-------------------------

Returns

A newly created record, or `NULL` on failure.

5.79.2.3 void ds_record_destroy (ds_record record)

Destroys a record and frees any associated resources.

Parameters

<i>record</i>	The record to destroy.
---------------	------------------------

5.79.2.4 void ds_record_destructor (void * record)

A record destructor function.

Parameters

<i>record</i>	The record to destroy.
---------------	------------------------

5.79.2.5 ds_str ds_record_get_field (ds_record record, const size_t index)

Retrieves the field at a specified index.

Parameters

<i>record</i>	The record from which to retrieve.
<i>index</i>	The index of the desired field.

Returns

A pointer to the field, or `NULL` if the index is out of range.

5.79.2.6 `ds_str ds_record_get_next_data (ds_record record)`

Returns the next field of the record.

This function returns the data of the "current field", and advances the current field pointer. Subsequent calls to this function will return successive fields.

Parameters

<i>record</i>	The record.
---------------	-------------

Returns

A pointer to the next field, or `NULL` if the end of the record has been reached.

5.79.2.7 `ds_str ds_record_make_delim_string (ds_record record, const char delim)`

Makes a delimited string from a record.

Parameters

<i>record</i>	The record.
<i>delim</i>	The delimiting character.

Returns

The delimited string, or `NULL` on failure.

5.79.2.8 `ds_str ds_record_make_values_string (ds_record record, enum ds_field_types * types)`

Makes a delimited SQL values string from a record.

Parameters

<i>record</i>	The record.
<i>types</i>	An array of types for each field, or <code>NULL</code> to assume they are all strings. The effect of this parameter is that string fields are quoted in the values string, whereas non-string fields are not.

Returns

The delimited values string, or `NULL` on failure.

5.79.2.9 `void ds_record_seek_start (ds_record record)`

Sets the current field to the first field of a record.

Parameters

<i>record</i>	The record.
---------------	-------------

5.79.2.10 void ds_record_set_field (ds_record *record*, const size_t *index*, ds_str *field*)

Sets a field of a record.

If the field is currently occupied, the existing field is `free()`d.

Parameters

<i>record</i>	The record to set.
<i>index</i>	The index of the field to set.
<i>field</i>	The value to which to set the field.

5.79.2.11 size_t ds_record_size (ds_record *record*)

Returns the size of a record.

Parameters

<i>record</i>	The record.
---------------	-------------

Returns

The size of the record.

5.79.2.12 ds_record ds_record_tokenize (ds_str *str*, const char *delim*)

Tokenizes a string into a record.

Parameters

<i>str</i>	The string to tokenize.
<i>delim</i>	The delimiting character.

Returns

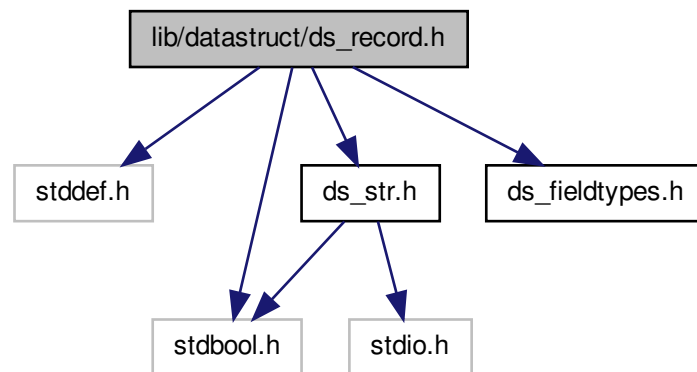
A new record containing the tokens.

5.80 lib/datastruct/ds_record.h File Reference

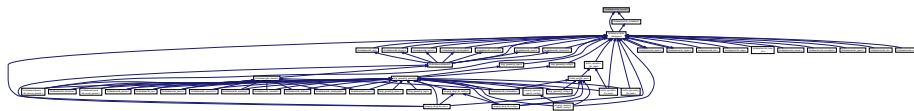
Interface to record data structure.

```
#include <stddef.h>
#include <stdbool.h>
#include "ds_str.h"
#include "ds_fieldtypes.h"
```

Include dependency graph for `ds_record.h`:



This graph shows which files directly or indirectly include this file:



Typedefs

- typedef struct `ds_record` * `ds_record`

Functions

- `ds_record ds_record_create` (const size_t size)
Creates a new record.
- void `ds_record_destroy` (`ds_record` record)
Destroys a record and frees any associated resources.
- void `ds_record_destructor` (void *record)
A record destructor function.
- void `ds_record_clear` (`ds_record` record)
Clears and `free()`s all the elements in a record.
- void `ds_record_set_field` (`ds_record` record, const size_t index, `ds_str` field)
Sets a field of a record.
- `ds_str ds_record_get_field` (`ds_record` record, const size_t index)
Retrieves the field at a specified index.
- size_t `ds_record_size` (`ds_record` record)
Returns the size of a record.
- void `ds_record_seek_start` (`ds_record` record)
Sets the current field to the first field of a record.
- `ds_str ds_record_get_next_data` (`ds_record` record)
Returns the next field of the record.

- [ds_record ds_record_tokenize](#) ([ds_str](#) str, const char delim)
Tokenizes a string into a record.
- [ds_str ds_record_make_delim_string](#) ([ds_record](#) record, const char delim)
Makes a delimited string from a record.
- [ds_str ds_record_make_values_string](#) ([ds_record](#) record, enum [ds_field_types](#) *types)
Makes a delimited SQL values string from a record.

5.80.1 Detailed Description

Interface to record data structure.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.80.2 Typedef Documentation

5.80.2.1 typedef struct [ds_record](#)* [ds_record](#)

Typedef for opaque record datatype

5.80.3 Function Documentation

5.80.3.1 void [ds_record_clear](#) ([ds_record](#) record)

Clears and `free()`s all the elements in a record.

Parameters

<i>record</i>	The record.
---------------	-------------

5.80.3.2 [ds_record](#) [ds_record_create](#) (const [size_t](#) size)

Creates a new record.

Parameters

<i>size</i>	The size of the record.
-------------	-------------------------

Returns

A newly created record, or `NULL` on failure.

5.80.3.3 void [ds_record_destroy](#) ([ds_record](#) record)

Destroys a record and frees any associated resources.

Parameters

<i>record</i>	The record to destroy.
---------------	------------------------

5.80.3.4 `void ds_record_destructor (void * record)`

A record destructor function.

Parameters

<i>record</i>	The record to destroy.
---------------	------------------------

5.80.3.5 `ds_str ds_record_get_field (ds_record record, const size_t index)`

Retrieves the field at a specified index.

Parameters

<i>record</i>	The record from which to retrieve.
<i>index</i>	The index of the desired field.

Returns

A pointer to the field, or `NULL` if the index is out of range.

5.80.3.6 `ds_str ds_record_get_next_data (ds_record record)`

Returns the next field of the record.

This function returns the data of the "current field", and advances the current field pointer. Subsequent calls to this function will return successive fields.

Parameters

<i>record</i>	The record.
---------------	-------------

Returns

A pointer to the next field, or `NULL` if the end of the record has been reached.

5.80.3.7 `ds_str ds_record_make_delim_string (ds_record record, const char delim)`

Makes a delimited string from a record.

Parameters

<i>record</i>	The record.
<i>delim</i>	The delimiting character.

Returns

The delimited string, or `NULL` on failure.

5.80.3.8 ds_str ds_record_make_values_string (ds_record record, enum ds_field_types * types)

Makes a delimited SQL values string from a record.

Parameters

<i>record</i>	The record.
<i>types</i>	An array of types for each field, or <code>NULL</code> to assume they are all strings. The effect of this parameter is that string fields are quoted in the values string, whereas non-string fields are not.

Returns

The delimited values string, or `NULL` on failure.

5.80.3.9 void ds_record_seek_start (ds_record record)

Sets the current field to the first field of a record.

Parameters

<i>record</i>	The record.
---------------	-------------

5.80.3.10 void ds_record_set_field (ds_record record, const size_t index, ds_str field)

Sets a field of a record.

If the field is currently occupied, the existing field is `free()`d.

Parameters

<i>record</i>	The record to set.
<i>index</i>	The index of the field to set.
<i>field</i>	The value to which to set the field.

5.80.3.11 size_t ds_record_size (ds_record record)

Returns the size of a record.

Parameters

<i>record</i>	The record.
---------------	-------------

Returns

The size of the record.

5.80.3.12 ds_record ds_record_tokenize (ds_str str, const char delim)

Tokenizes a string into a record.

Parameters

<i>str</i>	The string to tokenize.
<i>delim</i>	The delimiting character.

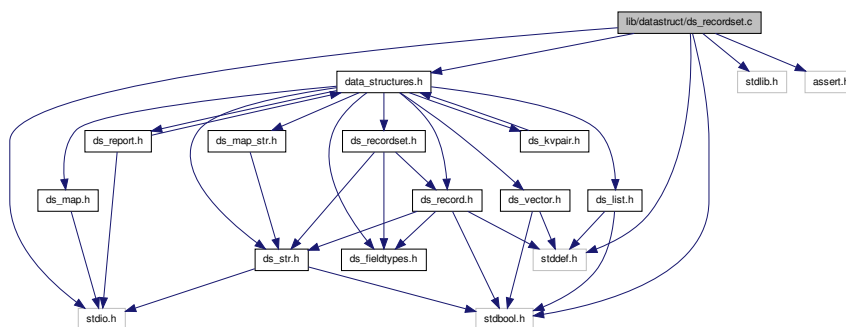
Returns

A new record containing the tokens.

5.81 lib/datastruct/ds_recordset.c File Reference

Implementation of query result set structure.

```
#include <stdio.h>
#include <stdlib.h>
#include <stddef.h>
#include <stdbool.h>
#include <assert.h>
#include "data_structures.h"
Include dependency graph for ds_recordset.c:
```



Data Structures

- struct [ds_recordset](#)

Functions

- [ds_recordset ds_recordset_create](#) (const size_t num_fields)
Creates a new record set.
- void [ds_recordset_destroy](#) (ds_recordset set)
Destroys a record set and frees associated resources.
- [ds_record ds_recordset_add_record](#) (ds_recordset set, ds_record record)
Adds a record to a record set.
- size_t [ds_recordset_num_fields](#) (ds_recordset set)
Returns the number of fields in a record set.
- size_t [ds_recordset_num_records](#) (ds_recordset set)
Returns the number of records in a record set.
- void [ds_recordset_set_headers](#) (ds_recordset set, ds_record headers)
Sets the record headers in a record set.
- void [ds_recordset_set_type](#) (ds_recordset set, const size_t index, const enum [ds_field_types](#) type)
Sets the type for a specified field.
- [ds_str ds_recordset_get_text_report](#) (ds_recordset set)
Returns a formatted text report for the record set.
- void [ds_recordset_seek_start](#) (ds_recordset set)

Sets the current record to the first record.

- `ds_record ds_recordset_next_record (ds_recordset set)`

Returns the next record in the record set.

- `ds_str ds_recordset_get_next_insert_query (ds_recordset set, const char *table_name)`

Gets the next SQL INSERT query.

5.81.1 Detailed Description

Implementation of query result set structure.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.81.2 Function Documentation

5.81.2.1 `ds_record ds_recordset_add_record (ds_recordset set, ds_record record)`

Adds a record to a record set.

The record *must* have the same number of fields as the number of fields provided to `ds_recordset_create()`.

Parameters

<i>set</i>	The record set to which to add.
<i>record</i>	The record to add.

Returns

A pointer to the new record (i.e. it returns the second parameter) or `NULL` on failure.

5.81.2.2 `ds_recordset ds_recordset_create (const size_t num_fields)`

Creates a new record set.

Parameters

<i>num_fields</i>	The non-zero number of fields in the record set.
-------------------	--

Returns

A pointer to the new record set.

5.81.2.3 `void ds_recordset_destroy (ds_recordset set)`

Destroys a record set and frees associated resources.

Parameters

<i>set</i>	The record set to destroy.
------------	----------------------------

5.81.2.4 ds_str ds_recordset_get_next_insert_query (ds_recordset set, const char * table_name)

Gets the next SQL INSERT query.

Parameters

<i>set</i>	The set.
<i>table_name</i>	The table name into which to insert.

Returns

The query. Caller is responsible for `free()` ing.

5.81.2.5 ds_str ds_recordset_get_text_report (ds_recordset set)

Returns a formatted text report for the record set.

The report is returned as a single multi-line string.

Parameters

<i>set</i>	The record set.
------------	-----------------

Returns

A pointer to the report. The caller is responsible for `free()` ing this pointer.

5.81.2.6 ds_record ds_recordset_next_record (ds_recordset set)

Returns the next record in the record set.

This function returns the "current record", and advances the current record pointer. Subsequent calls to this function will return successive records.

Parameters

<i>set</i>	The record set.
------------	-----------------

Returns

A pointer to the next record, or `NULL` if the end of the record set has been reached.

5.81.2.7 size_t ds_recordset_num_fields (ds_recordset set)

Returns the number of fields in a record set.

Parameters

<i>set</i>	The record set.
------------	-----------------

Returns

The number of fields in the record set.

5.81.2.8 `size_t ds_recordset_num_records (ds_recordset set)`

Returns the number of records in a record set.

Parameters

<i>set</i>	The record set.
------------	-----------------

Returns

The number of records in the record set.

5.81.2.9 `void ds_recordset_seek_start (ds_recordset set)`

Sets the current record to the first record.

Parameters

<i>set</i>	The record set.
------------	-----------------

5.81.2.10 `void ds_recordset_set_headers (ds_recordset set, ds_record headers)`

Sets the record headers in a record set.

Parameters

<i>set</i>	The record set.
<i>headers</i>	The headers, in the form of a ds_record of strings. The list <i>must</i> have the same number of elements as the number of fields provided to ds_recordset_create() .

5.81.2.11 `void ds_recordset_set_type (ds_recordset set, const size_t index, const enum ds_field_types type)`

Sets the type for a specified field.

Parameters

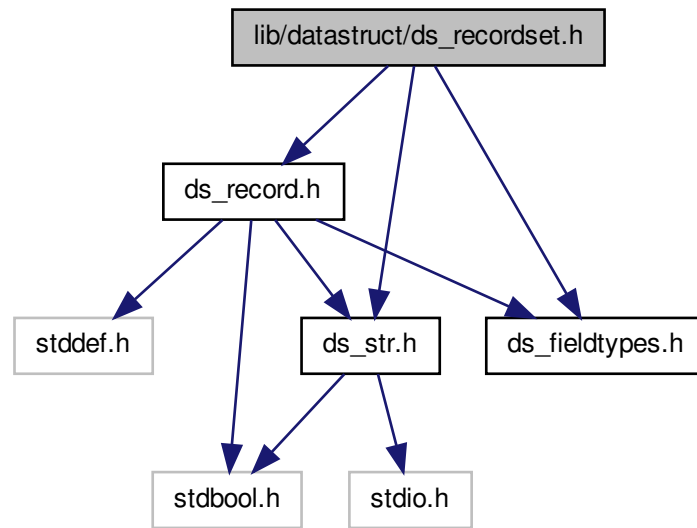
<i>set</i>	The record set.
<i>index</i>	The index to set.
<i>type</i>	The type for the field at the specified index.

5.82 lib/datastruct/ds_recordset.h File Reference

Interface to record set structure.

```
#include "ds_record.h"
#include "ds_str.h"
#include "ds_fieldtypes.h"
```

Include dependency graph for `ds_recordset.h`:



This graph shows which files directly or indirectly include this file:



Typedefs

- typedef struct `ds_recordset` * `ds_recordset`

Functions

- `ds_recordset ds_recordset_create` (const size_t num_fields)
Creates a new record set.
- void `ds_recordset_destroy` (`ds_recordset` set)
Destroys a record set and frees associated resources.
- `ds_record ds_recordset_add_record` (`ds_recordset` set, `ds_record` record)
Adds a record to a record set.
- size_t `ds_recordset_num_fields` (`ds_recordset` set)
Returns the number of fields in a record set.
- size_t `ds_recordset_num_records` (`ds_recordset` set)
Returns the number of records in a record set.
- void `ds_recordset_set_headers` (`ds_recordset` set, `ds_record` headers)
Sets the record headers in a record set.
- void `ds_recordset_set_type` (`ds_recordset` set, const size_t index, const enum `ds_field_types` type)
Sets the type for a specified field.

- [ds_str ds_recordset_get_text_report](#) ([ds_recordset](#) set)
Returns a formatted text report for the record set.
- [ds_str ds_recordset_get_next_insert_query](#) ([ds_recordset](#) set, const char *table_name)
Gets the next SQL INSERT query.
- void [ds_recordset_seek_start](#) ([ds_recordset](#) set)
Sets the current record to the first record.
- [ds_record ds_recordset_next_record](#) ([ds_recordset](#) set)
Returns the next record in the record set.

5.82.1 Detailed Description

Interface to record set structure.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.82.2 Typedef Documentation

5.82.2.1 typedef struct [ds_recordset*](#) [ds_recordset](#)

Typedef for opaque record set data type

5.82.3 Function Documentation

5.82.3.1 [ds_record ds_recordset_add_record](#) ([ds_recordset](#) set, [ds_record](#) record)

Adds a record to a record set.

The record *must* have the same number of fields as the number of fields provided to [ds_recordset_create](#) ().

Parameters

<i>set</i>	The record set to which to add.
<i>record</i>	The record to add.

Returns

A pointer to the new record (i.e. it returns the second parameter) or `NULL` on failure.

5.82.3.2 [ds_recordset ds_recordset_create](#) (const size_t num_fields)

Creates a new record set.

Parameters

<i>num_fields</i>	The non-zero number of fields in the record set.
-------------------	--

Returns

A pointer to the new record set.

5.82.3.3 void ds_recordset_destroy (ds_recordset set)

Destroys a record set and frees associated resources.

Parameters

<i>set</i>	The record set to destroy.
------------	----------------------------

5.82.3.4 ds_str ds_recordset_get_next_insert_query (ds_recordset set, const char * table_name)

Gets the next SQL INSERT query.

Parameters

<i>set</i>	The set.
<i>table_name</i>	The table name into which to insert.

Returns

The query. Caller is responsible for `free()` ing.

5.82.3.5 ds_str ds_recordset_get_text_report (ds_recordset set)

Returns a formatted text report for the record set.

The report is returned as a single multi-line string.

Parameters

<i>set</i>	The record set.
------------	-----------------

Returns

A pointer to the report. The caller is responsible for `free()` ing this pointer.

5.82.3.6 ds_record ds_recordset_next_record (ds_recordset set)

Returns the next record in the record set.

This function returns the "current record", and advances the current record pointer. Subsequent calls to this function will return successive records.

Parameters

<i>set</i>	The record set.
------------	-----------------

Returns

A pointer to the next record, or `NULL` if the end of the record set has been reached.

5.82.3.7 `size_t ds_recordset_num_fields (ds_recordset set)`

Returns the number of fields in a record set.

Parameters

<i>set</i>	The record set.
------------	-----------------

Returns

The number of fields in the record set.

5.82.3.8 `size_t ds_recordset_num_records (ds_recordset set)`

Returns the number of records in a record set.

Parameters

<i>set</i>	The record set.
------------	-----------------

Returns

The number of records in the record set.

5.82.3.9 `void ds_recordset_seek_start (ds_recordset set)`

Sets the current record to the first record.

Parameters

<i>set</i>	The record set.
------------	-----------------

5.82.3.10 `void ds_recordset_set_headers (ds_recordset set, ds_record headers)`

Sets the record headers in a record set.

Parameters

<i>set</i>	The record set.
<i>headers</i>	The headers, in the form of a ds_record of strings. The list <i>must</i> have the same number of elements as the number of fields provided to ds_recordset_create() .

5.82.3.11 `void ds_recordset_set_type (ds_recordset set, const size_t index, const enum ds_field_types type)`

Sets the type for a specified field.

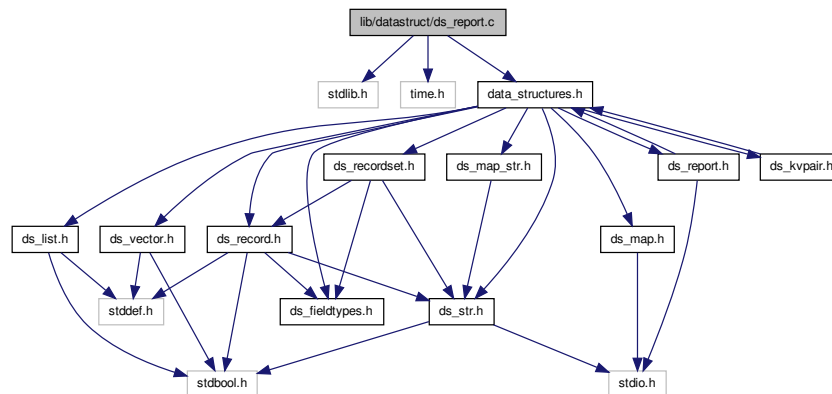
Parameters

<i>set</i>	The record set.
<i>index</i>	The index to set.
<i>type</i>	The type for the field at the specified index.

5.83 lib/datastruct/ds_report.c File Reference

Implementation of generic report data structure.

```
#include <stdlib.h>
#include <time.h>
#include "data_structures.h"
Include dependency graph for ds_report.c:
```



Data Structures

- struct [ds_report](#)

Macros

- `#define` [_XOPEN_SOURCE](#) 600

Functions

- [ds_report ds_report_create](#) (void)
Creates a new report.
- void [ds_report_destroy](#) (ds_report report)
Destroys a report and frees resources.
- void [ds_report_set_title](#) (ds_report report, ds_str title)
Sets the report title.
- void [ds_report_set_report_text](#) (ds_report report, ds_str report_text)
Sets the report text.
- bool [ds_report_add_header](#) (ds_report report, ds_str name, ds_str value)
Adds a header to the report.
- void [ds_report_print_text_report](#) (ds_report report, FILE *outfile)
Prints a text report to a file.

5.83.1 Detailed Description

Implementation of generic report data structure.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.83.2 Macro Definition Documentation**5.83.2.1 #define _XOPEN_SOURCE 600**

UNIX feature test macro

5.83.3 Function Documentation**5.83.3.1 bool ds_report_add_header (ds_report report, ds_str name, ds_str value)**

Adds a header to the report.

Parameters

<i>report</i>	The report.
<i>name</i>	The name of the header.
<i>value</i>	The value of the header.

Returns

`true` on success, `false` otherwise.

5.83.3.2 ds_report ds_report_create (void)

Creates a new report.

Returns

The new report, or `NULL` on failure.

5.83.3.3 void ds_report_destroy (ds_report report)

Destroys a report and frees resources.

Parameters

<i>report</i>	The report to destroy.
---------------	------------------------

5.83.3.4 void ds_report_print_text_report (ds_report report, FILE * outfile)

Prints a text report to a file.

Parameters

<i>report</i>	The report.
<i>outfile</i>	A pointer to the file to which to print.

5.83.3.5 void ds_report_set_report_text (ds_report report, ds_str report_text)

Sets the report text.

Parameters

<i>report</i>	The report.
<i>report_text</i>	The text. This string will be automatically freed when the report is destroyed, so you should not pass a string which you wish to retain.

5.83.3.6 void ds_report_set_title (ds_report report, ds_str title)

Sets the report title.

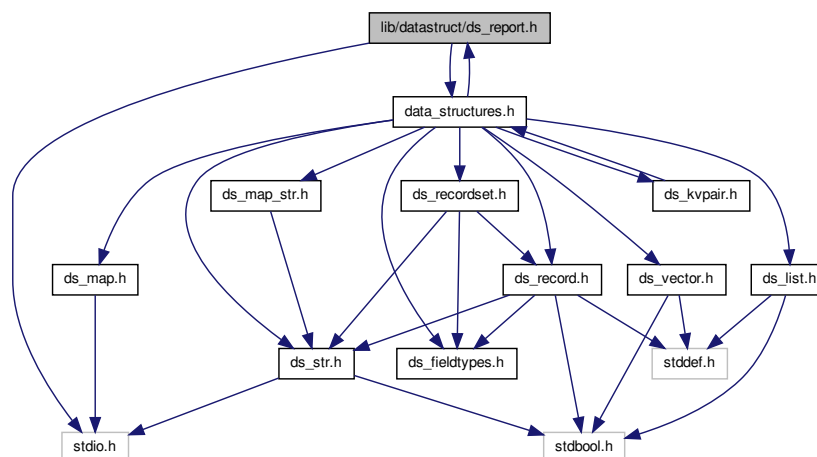
Parameters

<i>report</i>	The report.
<i>title</i>	The title. This string will be automatically freed when the report is destroyed, so you should not pass a string which you wish to retain.

5.84 lib/datastruct/ds_report.h File Reference

Interface to generic report data structure.

```
#include <stdio.h>
#include "data_structures.h"
Include dependency graph for ds_report.h:
```



This graph shows which files directly or indirectly include this file:



Typedefs

- typedef struct [ds_report](#) * [ds_report](#)

Functions

- [ds_report](#) [ds_report_create](#) (void)
Creates a new report.
- void [ds_report_destroy](#) ([ds_report](#) report)
Destroys a report and frees resources.
- void [ds_report_set_title](#) ([ds_report](#) report, [ds_str](#) title)
Sets the report title.
- void [ds_report_set_report_text](#) ([ds_report](#) report, [ds_str](#) report_text)
Sets the report text.
- void [ds_report_print_text_report](#) ([ds_report](#) report, FILE *outfile)
Prints a text report to a file.
- bool [ds_report_add_header](#) ([ds_report](#) report, [ds_str](#) name, [ds_str](#) value)
Adds a header to the report.

5.84.1 Detailed Description

Interface to generic report data structure.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.84.2 Typedef Documentation

5.84.2.1 typedef struct [ds_report](#)* [ds_report](#)

Opaque data type for report

5.84.3 Function Documentation

5.84.3.1 bool [ds_report_add_header](#) ([ds_report](#) report, [ds_str](#) name, [ds_str](#) value)

Adds a header to the report.

Parameters

<i>report</i>	The report.
<i>name</i>	The name of the header.
<i>value</i>	The value of the header.

Returns

`true` on success, `false` otherwise.

5.84.3.2 `ds_report ds_report_create (void)`

Creates a new report.

Returns

The new report, or `NULL` on failure.

5.84.3.3 `void ds_report_destroy (ds_report report)`

Destroys a report and frees resources.

Parameters

<i>report</i>	The report to destroy.
---------------	------------------------

5.84.3.4 `void ds_report_print_text_report (ds_report report, FILE * outfile)`

Prints a text report to a file.

Parameters

<i>report</i>	The report.
<i>outfile</i>	A pointer to the file to which to print.

5.84.3.5 `void ds_report_set_report_text (ds_report report, ds_str report_text)`

Sets the report text.

Parameters

<i>report</i>	The report.
<i>report_text</i>	The text. This string will be automatically freed when the report is destroyed, so you should not pass a string which you wish to retain.

5.84.3.6 `void ds_report_set_title (ds_report report, ds_str title)`

Sets the report title.

Parameters

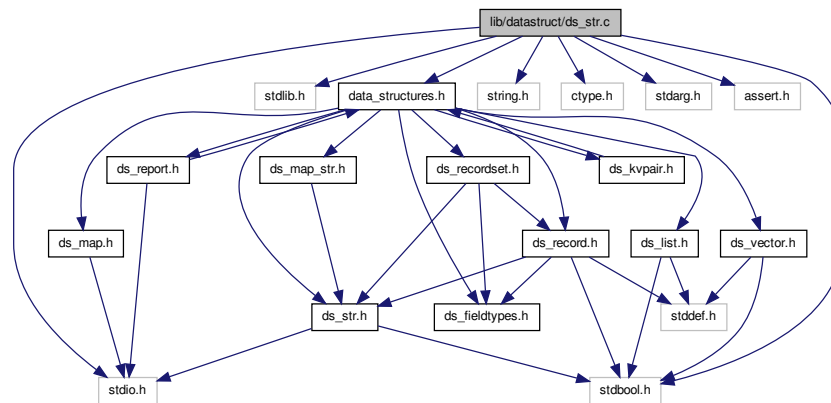
<i>report</i>	The report.
<i>title</i>	The title. This string will be automatically freed when the report is destroyed, so you should not pass a string which you wish to retain.

5.85 lib/datastruct/ds_str.c File Reference

Implementation of string data structure.

```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#include <string.h>
#include <ctype.h>
#include <stdarg.h>
#include <assert.h>
#include "data_structures.h"
```

Include dependency graph for ds_str.c:



Data Structures

- struct [ds_str](#)

Functions

- [ds_str ds_str_create_direct](#) (char *init_str, const size_t init_str_size)
Creates a string using allocated memory.
- [ds_str ds_str_create](#) (const char *init_str)
Creates a new string from a C-style string.
- [ds_str ds_str_dup](#) (ds_str src)
Creates a new string from another string.
- [ds_str ds_str_create_sprintf](#) (const char *format,...)
Creates a string with `sprintf()`-type format.
- void [ds_str_destroy](#) (ds_str str)
Destroys a string and releases allocated resources.
- void [ds_str_destructor](#) (void *str)
Destroys a string and releases allocated resources.
- [ds_str ds_str_assign](#) (ds_str dst, ds_str src)
Assigns a string to another.
- [ds_str ds_str_assign_cstr](#) (ds_str dst, const char *src)
Assigns a C-style string to a string.
- const char * [ds_str_cstr](#) (ds_str str)

- Returns a C-style string containing the string's contents.*

 - `size_t ds_str_length (ds_str str)`

Returns the length of a string.
 - `ds_str ds_str_size_to_fit (ds_str str)`

Reduces a string's capacity to fit its length.
 - `ds_str ds_str_concat (ds_str dst, ds_str src)`

Concatenates two strings.
 - `ds_str ds_str_concat_cstr (ds_str dst, const char *src)`

Concatenates a C-style string to a string.
 - `ds_str ds_str_trunc (ds_str str, const size_t length)`

Truncates a string.
 - `unsigned long ds_str_hash (ds_str str)`

Calculates a hash of a string.
 - `int ds_str_compare (ds_str s1, ds_str s2)`

Compares two strings.
 - `int ds_str_compare_cstr (ds_str s1, const char *s2)`

Compares a string with a C-style string.
 - `int ds_str_strchr (ds_str str, const char ch, const int start)`

Returns index of first occurrence of a character.
 - `ds_str ds_str_substr_left (ds_str str, const size_t numchars)`

Returns a left substring.
 - `ds_str ds_str_substr_right (ds_str str, const size_t numchars)`

Returns a right substring.
 - `void ds_str_split (ds_str src, ds_str *left, ds_str *right, const char sc)`

Splits a string.
 - `void ds_str_trim_leading (ds_str str)`

Trims leading whitespace in-place.
 - `void ds_str_trim_trailing (ds_str str)`

Trims trailing whitespace in-place.
 - `void ds_str_trim (ds_str str)`

Trims leading and trailing whitespace in-place.
 - `char ds_str_char_at_index (ds_str str, const size_t index)`

Returns the character at a specified index.
 - `bool ds_str_is_empty (ds_str str)`

Checks if a string is empty.
 - `bool ds_str_is_alnum (ds_str str)`

Checks if a string contains only alphanumeric characters.
 - `void ds_str_clear (ds_str str)`

Clears (empties) a string.
 - `bool ds_str_intval (ds_str str, const int base, int *value)`

Gets the integer value of a string.
 - `bool ds_str_doubleval (ds_str str, double *value)`

Gets the double value of a string.
 - `ds_str ds_str_getline (ds_str str, const size_t size, FILE *fp)`

Gets a line from a file and assigns it to a string.
 - `ds_str ds_str_decorate (ds_str str, ds_str left_dec, ds_str right_dec)`

Brackets a string with decoration strings.

5.85.1 Detailed Description

Implementation of string data structure.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.85.2 Function Documentation

5.85.2.1 `ds_str ds_str_assign (ds_str dst, ds_str src)`

Assigns a string to another.

Parameters

<i>dst</i>	The destination string.
<i>src</i>	The source string.

Returns

dst on success, `NULL` on failure.

5.85.2.2 `ds_str ds_str_assign_cstr (ds_str dst, const char * src)`

Assigns a C-style string to a string.

Parameters

<i>dst</i>	The destination string.
<i>src</i>	The source C-style string.

Returns

dst on success, `NULL` on failure.

5.85.2.3 `char ds_str_char_at_index (ds_str str, const size_t index)`

Returns the character at a specified index.

Parameters

<i>str</i>	The string.
<i>index</i>	The specified index.

Returns

The character at the specified index.

5.85.2.4 void ds_str_clear (ds_str str)

Clears (empties) a string.

Parameters

<i>str</i>	The string.
------------	-------------

5.85.2.5 int ds_str_compare (ds_str s1, ds_str s2)

Compares two strings.

Parameters

<i>s1</i>	The first string.
<i>s2</i>	The second string.

Returns

Less than, equal to, or greater than zero if s1 is found, respectively, to be less than, equal to, or greater than s2.

5.85.2.6 int ds_str_compare_cstr (ds_str s1, const char * s2)

Compares a string with a C-style string.

Parameters

<i>s1</i>	The first string.
<i>s2</i>	The second, C-Style string.

Returns

Less than, equal to, or greater than zero if s1 is found, respectively, to be less than, equal to, or greater than s2.

5.85.2.7 ds_str ds_str_concat (ds_str dst, ds_str src)

Concatenates two strings.

Parameters

<i>dst</i>	The destination string.
<i>src</i>	The source strings.

Returns

The destination string, or NULL on failure.

5.85.2.8 ds_str ds_str_concat_cstr (ds_str dst, const char * src)

Concatenates a C-style string to a string.

Parameters

<i>dst</i>	The destination string.
<i>src</i>	The source strings.

Returns

The destination string, or `NULL` on failure.

5.85.2.9 ds_str ds_str_create (const char * *init_str*)

Creates a new string from a C-style string.

Parameters

<i>init_str</i>	The C-style string.
-----------------	---------------------

Returns

The new string, or `NULL` on failure.

5.85.2.10 ds_str ds_str_create_direct (char * *init_str*, const size_t *init_str_size*)

Creates a string using allocated memory.

The normal construction functions duplicate the string used to create it. In cases where allocated memory is already available (e.g. in `ds_str_create_sprintf()`) this function allows that memory to be directly assigned to the string, avoiding an unnecessary duplication.

Parameters

<i>init_str</i>	The allocated memory. IMPORTANT: If the construction of the string fails, this memory will be <code>free()</code> d.
<i>init_str_size</i>	The size of the allocated memory. IMPORTANT: The string's length is assumed to be one less than this quantity, and a call to <code>strlen()</code> is NOT performed.

Returns

The new string, or `NULL` on failure.

5.85.2.11 ds_str ds_str_create_sprintf (const char * *format*, ...)

Creates a string with `sprintf()`-type format.

Parameters

<i>format</i>	The format string.
...	The subsequent arguments as specified by the format string.

Returns

The new string, or `NULL` on failure.

5.85.2.12 `const char* ds_str_cstr (ds_str str)`

Returns a C-style string containing the string's contents.

Parameters

<i>str</i>	The string.
------------	-------------

Returns

The C-style string containing the string's contents. The caller should not directly modify this string.

5.85.2.13 `ds_str ds_str_decorate (ds_str str, ds_str left_dec, ds_str right_dec)`

Brackets a string with decoration strings.

Parameters

<i>str</i>	The string to decorate.
<i>left_dec</i>	The string to add to the left of <i>str</i> .
<i>right_dec</i>	The string to add to the right of <i>str</i> , or NULL to add <i>left_dec</i> to both sides.

Returns

The decorated string.

5.85.2.14 `void ds_str_destroy (ds_str str)`

Destroys a string and releases allocated resources.

Parameters

<i>str</i>	The string to destroy..
------------	-------------------------

5.85.2.15 `void ds_str_destructor (void * str)`

Destroys a string and releases allocated resources.

This function calls `ds_str_destroy()`, and can be passed to a data structure expecting a destructor function with the signature `void (*)(void *)`.

Parameters

<i>str</i>	The string to destroy.
------------	------------------------

5.85.2.16 `bool ds_str_doubleval (ds_str str, double * value)`

Gets the double value of a string.

Parameters

<i>str</i>	The string.
<i>value</i>	A pointer to the double in which to store the value. Zero is stored if the string does not contain a valid double value.

Returns

`true` on successful conversion, `false` if the string does not contain a valid double value.

5.85.2.17 ds_str ds_str_dup (ds_str src)

Creates a new string from another string.

Parameters

<i>src</i>	The other string.
------------	-------------------

Returns

The new string, or `NULL` on failure.

5.85.2.18 ds_str ds_str_getline (ds_str str, const size_t size, FILE * fp)

Gets a line from a file and assigns it to a string.

Any trailing newline character is stripped.

Parameters

<i>str</i>	The string.
<i>size</i>	The maximum number of bytes to read, including the null.
<i>fp</i>	The file pointer from which to read.

Returns

`dst`

5.85.2.19 unsigned long ds_str_hash (ds_str str)

Calculates a hash of a string.

Uses Dan Bernstein's djb2 algorithm.

Parameters

<i>str</i>	The string.
------------	-------------

Returns

The hash value

5.85.2.20 bool ds_str_intval (ds_str str, const int base, int * value)

Gets the integer value of a string.

Parameters

<i>str</i>	The string.
<i>base</i>	The base of the integer. This has the same meaning as the third argument to standard C <code>strtol()</code> .

<i>value</i>	A pointer to the integer in which to store the value. Zero is stored if the string does not contain a valid integer value.
--------------	--

Returns

`true` on successful conversion, `false` if the string does not contain a valid integer value.

5.85.2.21 `bool ds_str_is_alnum (ds_str str)`

Checks if a string contains only alphanumeric characters.

The string must contain *some* alphanumeric characters to check `true`, i.e. the string must be non-empty. Thus it can be used to check that a string does indeed contain content, and that that content is solely alphanumeric.

Parameters

<i>str</i>	The string.
------------	-------------

Returns

`true` if the string contains only alphanumeric characters, `false` otherwise.

5.85.2.22 `bool ds_str_is_empty (ds_str str)`

Checks if a string is empty.

Parameters

<i>str</i>	The string.
------------	-------------

Returns

`true` if the string is empty, `false` otherwise.

5.85.2.23 `size_t ds_str_length (ds_str str)`

Returns the length of a string.

Parameters

<i>str</i>	The string.
------------	-------------

Returns

The length of the string.

5.85.2.24 `ds_str ds_str_size_to_fit (ds_str str)`

Reduces a string's capacity to fit its length.

Parameters

<i>str</i>	The string to size.
------------	---------------------

Returns

str, or NULL on failure.

5.85.2.25 void ds_str_split (ds_str *src*, ds_str * *left*, ds_str * *right*, const char *sc*)

Splits a string.

Parameters

<i>src</i>	The string to split.
<i>left</i>	Pointer to left substring (modified)
<i>right</i>	Pointer to right substring (modified)
<i>sc</i>	Split character.

5.85.2.26 int ds_str_strchr (ds_str *str*, const char *ch*, const int *start*)

Returns index of first occurrence of a character.

Parameters

<i>str</i>	The string.
<i>ch</i>	The character for which to search.
<i>start</i>	The index of the string at which to start looking. Set this to non-zero to begin searching from a point other than the first character of the string.

Returns

The index of the first occurrence, or -1 if the character was not found.

5.85.2.27 ds_str ds_str_substr_left (ds_str *str*, const size_t *numchars*)

Returns a left substring.

Parameters

<i>str</i>	The string.
<i>numchars</i>	The number of left characters to return. If this is greater than the length of the string, the whole string is returned.

Returns

A new string representing the substring.

5.85.2.28 ds_str ds_str_substr_right (ds_str *str*, const size_t *numchars*)

Returns a right substring.

Parameters

<i>str</i>	The string.
<i>numchars</i>	The number of right characters to return. If this is greater than the length of the string, the whole string is returned.

Returns

A new string representing the substring.

5.85.2.29 void ds_str_trim (ds_str str)

Trims leading and trailing whitespace in-place.

Parameters

<i>str</i>	The string.
------------	-------------

5.85.2.30 void ds_str_trim.leading (ds_str str)

Trims leading whitespace in-place.

Parameters

<i>str</i>	The string.
------------	-------------

5.85.2.31 void ds_str_trim.trailing (ds_str str)

Trims trailing whitespace in-place.

Parameters

<i>str</i>	The string.
------------	-------------

5.85.2.32 ds_str ds_str_trunc (ds_str str, const size_t length)

Truncates a string.

Parameters

<i>str</i>	The string.
<i>length</i>	The new length to which to truncate.

Returns

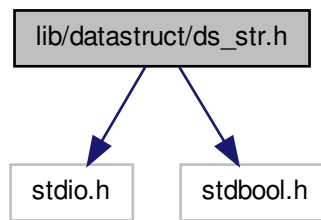
The original string, or `NULL` on failure.

5.86 lib/datastruct/ds_str.h File Reference

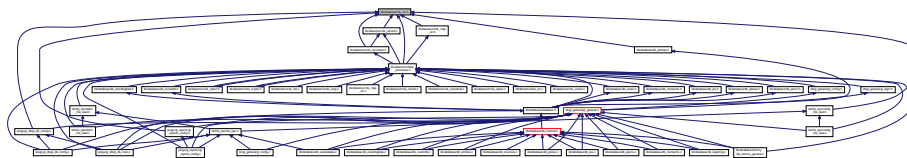
Interface to string data structure.

```
#include <stdio.h>
#include <stdbool.h>
```

Include dependency graph for ds_str.h:



This graph shows which files directly or indirectly include this file:



Typedefs

- typedef struct [ds_str](#) * [ds_str](#)

Functions

- [ds_str ds_str_create](#) (const char *init_str)
Creates a new string from a C-style string.
- [ds_str ds_str_dup](#) (ds_str src)
Creates a new string from another string.
- [ds_str ds_str_create_sprintf](#) (const char *format,...)
Creates a string with `sprintf()`-type format.
- [ds_str ds_str_create_direct](#) (char *init_str, const size_t init_str_size)
Creates a string using allocated memory.
- void [ds_str_destroy](#) (ds_str str)
Destroys a string and releases allocated resources.
- void [ds_str_destructor](#) (void *str)
Destroys a string and releases allocated resources.
- [ds_str ds_str_assign](#) (ds_str dst, ds_str src)
Assigns a string to another.
- [ds_str ds_str_assign_cstr](#) (ds_str dst, const char *src)
Assigns a C-style string to a string.
- const char * [ds_str_cstr](#) (ds_str str)
Returns a C-style string containing the string's contents.
- size_t [ds_str_length](#) (ds_str str)
Returns the length of a string.

- [ds_str ds_str_size_to_fit](#) ([ds_str](#) str)
Reduces a string's capacity to fit its length.
- [ds_str ds_str_concat](#) ([ds_str](#) dst, [ds_str](#) src)
Concatenates two strings.
- [ds_str ds_str_concat_cstr](#) ([ds_str](#) dst, const char *src)
Concatenates a C-style string to a string.
- [ds_str ds_str_trunc](#) ([ds_str](#) str, const size_t length)
Truncates a string.
- unsigned long [ds_str_hash](#) ([ds_str](#) str)
Calculates a hash of a string.
- int [ds_str_compare](#) ([ds_str](#) s1, [ds_str](#) s2)
Compares two strings.
- int [ds_str_compare_cstr](#) ([ds_str](#) s1, const char *s2)
Compares a string with a C-style string.
- int [ds_str_strchr](#) ([ds_str](#) str, const char ch, const int start)
Returns index of first occurrence of a character.
- [ds_str ds_str_substr_left](#) ([ds_str](#) str, const size_t numchars)
Returns a left substring.
- [ds_str ds_str_substr_right](#) ([ds_str](#) str, const size_t numchars)
Returns a right substring.
- void [ds_str_split](#) ([ds_str](#) src, [ds_str](#) *left, [ds_str](#) *right, const char sc)
Splits a string.
- void [ds_str_trim_leading](#) ([ds_str](#) str)
Trims leading whitespace in-place.
- void [ds_str_trim_trailing](#) ([ds_str](#) str)
Trims trailing whitespace in-place.
- void [ds_str_trim](#) ([ds_str](#) str)
Trims leading and trailing whitespace in-place.
- char [ds_str_char_at_index](#) ([ds_str](#) str, const size_t index)
Returns the character at a specified index.
- bool [ds_str_is_empty](#) ([ds_str](#) str)
Checks if a string is empty.
- bool [ds_str_is_alnum](#) ([ds_str](#) str)
Checks if a string contains only alphanumeric characters.
- void [ds_str_clear](#) ([ds_str](#) str)
Clears (empties) a string.
- bool [ds_str_intval](#) ([ds_str](#) str, const int base, int *value)
Gets the integer value of a string.
- bool [ds_str_doubleval](#) ([ds_str](#) str, double *value)
Gets the double value of a string.
- [ds_str ds_str_getline](#) ([ds_str](#) str, const size_t size, FILE *fp)
Gets a line from a file and assigns it to a string.
- [ds_str ds_str_decorate](#) ([ds_str](#) str, [ds_str](#) left_dec, [ds_str](#) right_dec)
Brackets a string with decoration strings.

5.86.1 Detailed Description

Interface to string data structure.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.86.2 Typedef Documentation

5.86.2.1 typedef struct ds_str* ds_str

Opaque data type for string

5.86.3 Function Documentation

5.86.3.1 ds_str ds_str_assign (ds_str dst, ds_str src)

Assigns a string to another.

Parameters

<i>dst</i>	The destination string.
<i>src</i>	The source string.

Returns

dst on success, NULL on failure.

5.86.3.2 ds_str ds_str_assign_cstr (ds_str dst, const char * src)

Assigns a C-style string to a string.

Parameters

<i>dst</i>	The destination string.
<i>src</i>	The source C-style string.

Returns

dst on success, NULL on failure.

5.86.3.3 char ds_str_char_at_index (ds_str str, const size_t index)

Returns the character at a specified index.

Parameters

<i>str</i>	The string.
<i>index</i>	The specified index.

Returns

The character at the specified index.

5.86.3.4 void ds_str_clear (ds_str str)

Clears (empties) a string.

Parameters

<i>str</i>	The string.
------------	-------------

5.86.3.5 int ds_str_compare (ds_str s1, ds_str s2)

Compares two strings.

Parameters

<i>s1</i>	The first string.
<i>s2</i>	The second string.

Returns

Less than, equal to, or greater than zero if *s1* is found, respectively, to be less than, equal to, or greater than *s2*.

5.86.3.6 int ds_str_compare_cstr (ds_str s1, const char * s2)

Compares a string with a C-style string.

Parameters

<i>s1</i>	The first string.
<i>s2</i>	The second, C-Style string.

Returns

Less than, equal to, or greater than zero if *s1* is found, respectively, to be less than, equal to, or greater than *s2*.

5.86.3.7 ds_str ds_str_concat (ds_str dst, ds_str src)

Concatenates two strings.

Parameters

<i>dst</i>	The destination string.
<i>src</i>	The source strings.

Returns

The destination string, or `NULL` on failure.

5.86.3.8 ds_str ds_str_concat_cstr (ds_str dst, const char * src)

Concatenates a C-style string to a string.

Parameters

<i>dst</i>	The destination string.
<i>src</i>	The source strings.

Returns

The destination string, or `NULL` on failure.

5.86.3.9 ds_str ds_str_create (const char * init_str)

Creates a new string from a C-style string.

Parameters

<i>init_str</i>	The C-style string.
-----------------	---------------------

Returns

The new string, or `NULL` on failure.

5.86.3.10 ds_str ds_str_create_direct (char * init_str, const size_t init_str_size)

Creates a string using allocated memory.

The normal construction functions duplicate the string used to create it. In cases where allocated memory is already available (e.g. in `ds_str_create_sprintf()`) this function allows that memory to be directly assigned to the string, avoiding an unnecessary duplication.

Parameters

<i>init_str</i>	The allocated memory. IMPORTANT: If the construction of the string fails, this memory will be <code>free()</code> d.
<i>init_str_size</i>	The size of the allocated memory. IMPORTANT: The string's length is assumed to be one less than this quantity, and a call to <code>strlen()</code> is NOT performed.

Returns

The new string, or `NULL` on failure.

5.86.3.11 ds_str ds_str_create_sprintf (const char * format, ...)

Creates a string with `sprintf()`-type format.

Parameters

<i>format</i>	The format string.
<i>...</i>	The subsequent arguments as specified by the format string.

Returns

The new string, or `NULL` on failure.

5.86.3.12 `const char* ds_str_cstr (ds_str str)`

Returns a C-style string containing the string's contents.

Parameters

<i>str</i>	The string.
------------	-------------

Returns

The C-style string containing the string's contents. The caller should not directly modify this string.

5.86.3.13 `ds_str ds_str_decorate (ds_str str, ds_str left_dec, ds_str right_dec)`

Brackets a string with decoration strings.

Parameters

<i>str</i>	The string to decorate.
<i>left_dec</i>	The string to add to the left of <i>str</i> .
<i>right_dec</i>	The string to add to the right of <i>str</i> , or <code>NULL</code> to add <i>left_dec</i> to both sides.

Returns

The decorated string.

5.86.3.14 `void ds_str_destroy (ds_str str)`

Destroys a string and releases allocated resources.

Parameters

<i>str</i>	The string to destroy..
------------	-------------------------

5.86.3.15 `void ds_str_destructor (void * str)`

Destroys a string and releases allocated resources.

This function calls `ds_str_destroy()`, and can be passed to a data structure expecting a destructor function with the signature `void (*)(void *)`.

Parameters

<i>str</i>	The string to destroy.
------------	------------------------

5.86.3.16 `bool ds_str_doubleval (ds_str str, double * value)`

Gets the double value of a string.

Parameters

<i>str</i>	The string.
<i>value</i>	A pointer to the double in which to store the value. Zero is stored if the string does not contain a valid double value.

Returns

`true` on successful conversion, `false` if the string does not contain a valid double value.

5.86.3.17 ds_str ds_str_dup (ds_str src)

Creates a new string from another string.

Parameters

<i>src</i>	The other string.
------------	-------------------

Returns

The new string, or `NULL` on failure.

5.86.3.18 ds_str ds_str_getline (ds_str str, const size_t size, FILE * fp)

Gets a line from a file and assigns it to a string.

Any trailing newline character is stripped.

Parameters

<i>str</i>	The string.
<i>size</i>	The maximum number of bytes to read, including the null.
<i>fp</i>	The file pointer from which to read.

Returns

`dst`

5.86.3.19 unsigned long ds_str_hash (ds_str str)

Calculates a hash of a string.

Uses Dan Bernstein's djb2 algorithm.

Parameters

<i>str</i>	The string.
------------	-------------

Returns

The hash value

5.86.3.20 bool ds_str_intval (ds_str str, const int base, int * value)

Gets the integer value of a string.

Parameters

<i>str</i>	The string.
<i>base</i>	The base of the integer. This has the same meaning as the third argument to standard C <code>strtol()</code> .
<i>value</i>	A pointer to the integer in which to store the value. Zero is stored if the string does not contain a valid integer value.

Returns

`true` on successful conversion, `false` if the string does not contain a valid integer value.

5.86.3.21 `bool ds_str_is_alnum (ds_str str)`

Checks if a string contains only alphanumeric characters.

The string must contain *some* alphanumeric characters to check `true`, i.e. the string must be non-empty. Thus it can be used to check that a string does indeed contain content, and that that content is solely alphanumeric.

Parameters

<i>str</i>	The string.
------------	-------------

Returns

`true` if the string contains only alphanumeric characters, `false` otherwise.

5.86.3.22 `bool ds_str_is_empty (ds_str str)`

Checks if a string is empty.

Parameters

<i>str</i>	The string.
------------	-------------

Returns

`true` if the string is empty, `false` otherwise.

5.86.3.23 `size_t ds_str_length (ds_str str)`

Returns the length of a string.

Parameters

<i>str</i>	The string.
------------	-------------

Returns

The length of the string.

5.86.3.24 `ds_str ds_str_size_to_fit (ds_str str)`

Reduces a string's capacity to fit its length.

Parameters

<i>str</i>	The string to size.
------------	---------------------

Returns

str, or NULL on failure.

5.86.3.25 void ds_str_split (ds_str *src*, ds_str * *left*, ds_str * *right*, const char *sc*)

Splits a string.

Parameters

<i>src</i>	The string to split.
<i>left</i>	Pointer to left substring (modified)
<i>right</i>	Pointer to right substring (modified)
<i>sc</i>	Split character.

5.86.3.26 int ds_str_strchr (ds_str *str*, const char *ch*, const int *start*)

Returns index of first occurrence of a character.

Parameters

<i>str</i>	The string.
<i>ch</i>	The character for which to search.
<i>start</i>	The index of the string at which to start looking. Set this to non-zero to begin searching from a point other than the first character of the string.

Returns

The index of the first occurrence, or -1 if the character was not found.

5.86.3.27 ds_str ds_str_substr_left (ds_str *str*, const size_t *numchars*)

Returns a left substring.

Parameters

<i>str</i>	The string.
<i>numchars</i>	The number of left characters to return. If this is greater than the length of the string, the whole string is returned.

Returns

A new string representing the substring.

5.86.3.28 ds_str ds_str_substr_right (ds_str *str*, const size_t *numchars*)

Returns a right substring.

Parameters

<i>str</i>	The string.
<i>numchars</i>	The number of right characters to return. If this is greater than the length of the string, the whole string is returned.

Returns

A new string representing the substring.

5.86.3.29 void ds_str_trim (ds_str str)

Trims leading and trailing whitespace in-place.

Parameters

<i>str</i>	The string.
------------	-------------

5.86.3.30 void ds_str_trim.leading (ds_str str)

Trims leading whitespace in-place.

Parameters

<i>str</i>	The string.
------------	-------------

5.86.3.31 void ds_str_trim.trailing (ds_str str)

Trims trailing whitespace in-place.

Parameters

<i>str</i>	The string.
------------	-------------

5.86.3.32 ds_str ds_str_trunc (ds_str str, const size_t length)

Truncates a string.

Parameters

<i>str</i>	The string.
<i>length</i>	The new length to which to truncate.

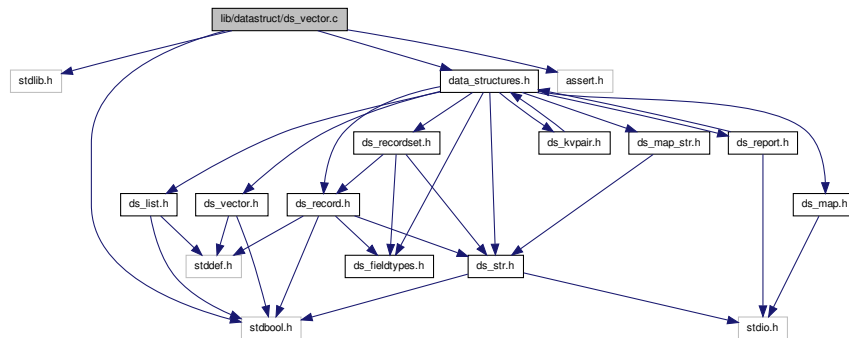
Returns

The original string, or `NULL` on failure.

5.87 lib/datastruct/ds_vector.c File Reference

Implementation of generic doubly-linked vector data structure.

```
#include <stdlib.h>
#include <stdbool.h>
#include <assert.h>
#include "data_structures.h"
Include dependency graph for ds_vector.c:
```



Data Structures

- struct [ds_vector](#)

Functions

- [ds_vector ds_vector_create](#) (const size_t size, const bool free_on_delete, void(*destructor)(void *))
Creates a new vector.
- void [ds_vector_destroy](#) (ds_vector vector)
Destroys a vector and frees any associated resources.
- void [ds_vector_destructor](#) (void *vector)
A vector destructor function.
- void [ds_vector_clear](#) (ds_vector vector)
Clears all the elements in a vector.
- void [ds_vector_set](#) (ds_vector vector, const size_t index, void *element)
Sets an element of a vector.
- void * [ds_vector_element](#) (ds_vector vector, const size_t index)
Retrieves the data at a specified index.
- size_t [ds_vector_size](#) (ds_vector vector)
Returns the size of a vector.
- void [ds_vector_seek_start](#) (ds_vector vector)
Sets the current element to the first element of a vector.
- void * [ds_vector_get_next_data](#) (ds_vector vector)
Returns the next element of the vector.

5.87.1 Detailed Description

Implementation of generic doubly-linked vector data structure.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.87.2 Function Documentation

5.87.2.1 void ds_vector_clear (ds_vector vector)

Clears all the elements in a vector.

If the vector was created with `free_on_delete`, the elements are `free()`d prior to being cleared (i.e. set to `NULL`).

Parameters

<i>vector</i>	The vector.
---------------	-------------

5.87.2.2 ds_vector ds_vector_create (const size_t size, const bool free_on_delete, void(*) (void *) destructor)

Creates a new vector.

Parameters

<i>size</i>	The size of the vector.
<i>free_on_delete</i>	Set to <code>true</code> if the vector elements should be destroyed when removed from the vector, and when the vector itself is destroyed. If set to <code>false</code> , the caller is responsible for destroying the elements prior to destroying the vector.
<i>destructor</i>	Pointer to a destructor function to use for destroying the vector elements, when <code>free_on_delete</code> is true. If this is set to <code>NULL</code> , <code>free()</code> from the standard C library will be used to destroy the elements.

Returns

A newly created vector, or `NULL` on failure.

5.87.2.3 void ds_vector_destroy (ds_vector vector)

Destroys a vector and frees any associated resources.

Parameters

<i>vector</i>	The vector to destroy.
---------------	------------------------

5.87.2.4 void ds_vector_destructor (void * vector)

A vector destructor function.

This function may be passed to `ds_vector_create()` when creating a vector of vectors. It calls `ds_vector_destroy()`, but the parameter of `ds_vector_destroy()` is not compatible with the function signature expected by `ds_vector_create()`, so this function provides an appropriate interface.

Parameters

<i>vector</i>	The vector to destroy.
---------------	------------------------

5.87.2.5 void* ds_vector_element (ds_vector vector, const size_t index)

Retrieves the data at a specified index.

Parameters

<i>vector</i>	The vector from which to retrieve.
<i>index</i>	The index of the desired element.

Returns

A pointer to the data, or `NULL` if the index is out of range.

5.87.2.6 void* ds_vector_get_next_data (ds_vector vector)

Returns the next element of the vector.

This function returns the data of the "current element", and advances the current element pointer. Subsequent calls to this function will return successive elements.

Parameters

<i>vector</i>	The vector.
---------------	-------------

Returns

A pointer to the next element, or `NULL` if the end of the vector has been reached.

5.87.2.7 void ds_vector_seek_start (ds_vector vector)

Sets the current element to the first element of a vector.

Parameters

<i>vector</i>	The vector.
---------------	-------------

5.87.2.8 void ds_vector_set (ds_vector vector, const size_t index, void * element)

Sets an element of a vector.

If the element is currently occupied, the existing element is `free()`d.

Parameters

<i>vector</i>	The vector to which to set.
<i>index</i>	The index of the element to set.
<i>element</i>	The element to set.

5.87.2.9 size_t ds_vector_size (ds_vector vector)

Returns the size of a vector.

Parameters

<i>vector</i>	The vector.
---------------	-------------

Returns

The size of the vector.

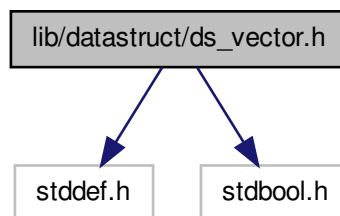
5.88 lib/datastruct/ds_vector.h File Reference

Interface to generic doubly-linked vector data structure.

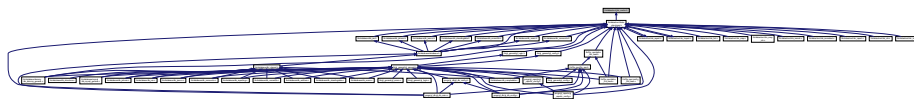
```
#include <stddef.h>
```

```
#include <stdbool.h>
```

Include dependency graph for ds_vector.h:



This graph shows which files directly or indirectly include this file:



Typedefs

- typedef struct [ds_vector](#) * [ds_vector](#)

Functions

- [ds_vector ds_vector_create](#) (const size_t size, const bool free_on_delete, void(*destructor)(void *))
Creates a new vector.
- void [ds_vector_destroy](#) ([ds_vector](#) vector)
Destroys a vector and frees any associated resources.
- void [ds_vector_destructor](#) (void *vector)
A vector destructor function.
- void [ds_vector_clear](#) ([ds_vector](#) vector)
Clears all the elements in a vector.
- void [ds_vector_set](#) ([ds_vector](#) vector, const size_t index, void *element)
Sets an element of a vector.

- void * [ds_vector_element](#) ([ds_vector](#) vector, const size_t index)
Retrieves the data at a specified index.
- size_t [ds_vector_size](#) ([ds_vector](#) vector)
Returns the size of a vector.
- void [ds_vector_seek_start](#) ([ds_vector](#) vector)
Sets the current element to the first element of a vector.
- void * [ds_vector_get_next_data](#) ([ds_vector](#) vector)
Returns the next element of the vector.

5.88.1 Detailed Description

Interface to generic doubly-linked vector data structure.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.88.2 Typedef Documentation

5.88.2.1 typedef struct ds_vector* ds_vector

Typedef for opaque vector datatype

5.88.3 Function Documentation

5.88.3.1 void ds_vector_clear (ds_vector vector)

Clears all the elements in a vector.

If the vector was created with `free_on_delete`, the elements are `free()`d prior to being cleared (i.e. set to `NULL`).

Parameters

<i>vector</i>	The vector.
---------------	-------------

5.88.3.2 ds_vector ds_vector_create (const size_t size, const bool free_on_delete, void(*) (void *) destructor)

Creates a new vector.

Parameters

<i>size</i>	The size of the vector.
<i>free_on_delete</i>	Set to <code>true</code> if the vector elements should be destroyed when removed from the vector, and when the vector itself is destroyed. If set to <code>false</code> , the caller is responsible for destroying the elements prior to destroying the vector.
<i>destructor</i>	Pointer to a destructor function to use for destroying the vector elements, when <code>free_on_delete</code> is true. If this is set to <code>NULL</code> , <code>free()</code> from the standard C library will be used to destroy the elements.

Returns

A newly created vector, or `NULL` on failure.

5.88.3.3 void ds_vector_destroy (ds_vector vector)

Destroys a vector and frees any associated resources.

Parameters

<i>vector</i>	The vector to destroy.
---------------	------------------------

5.88.3.4 void ds_vector_destructor (void * vector)

A vector destructor function.

This function may be passed to `ds_vector_create()` when creating a vector of vectors. It calls `ds_vector_destroy()`, but the parameter of `ds_vector_destroy()` is not compatible with the function signature expected by `ds_vector_create()`, so this function provides an appropriate interface.

Parameters

<i>vector</i>	The vector to destroy.
---------------	------------------------

5.88.3.5 void* ds_vector_element (ds_vector vector, const size_t index)

Retrieves the data at a specified index.

Parameters

<i>vector</i>	The vector from which to retrieve.
<i>index</i>	The index of the desired element.

Returns

A pointer to the data, or `NULL` if the index is out of range.

5.88.3.6 void* ds_vector_get_next_data (ds_vector vector)

Returns the next element of the vector.

This function returns the data of the "current element", and advances the current element pointer. Subsequent calls to this function will return successive elements.

Parameters

<i>vector</i>	The vector.
---------------	-------------

Returns

A pointer to the next element, or `NULL` if the end of the vector has been reached.

5.88.3.7 void ds_vector_seek_start (ds_vector vector)

Sets the current element to the first element of a vector.

Parameters

<i>vector</i>	The vector.
---------------	-------------

5.88.3.8 void ds_vector_set (ds_vector vector, const size_t index, void * element)

Sets an element of a vector.

If the element is currently occupied, the existing element is `free()`d.

Parameters

<i>vector</i>	The vector to which to set.
<i>index</i>	The index of the element to set.
<i>element</i>	The element to set.

5.88.3.9 size_t ds_vector_size (ds_vector vector)

Returns the size of a vector.

Parameters

<i>vector</i>	The vector.
---------------	-------------

Returns

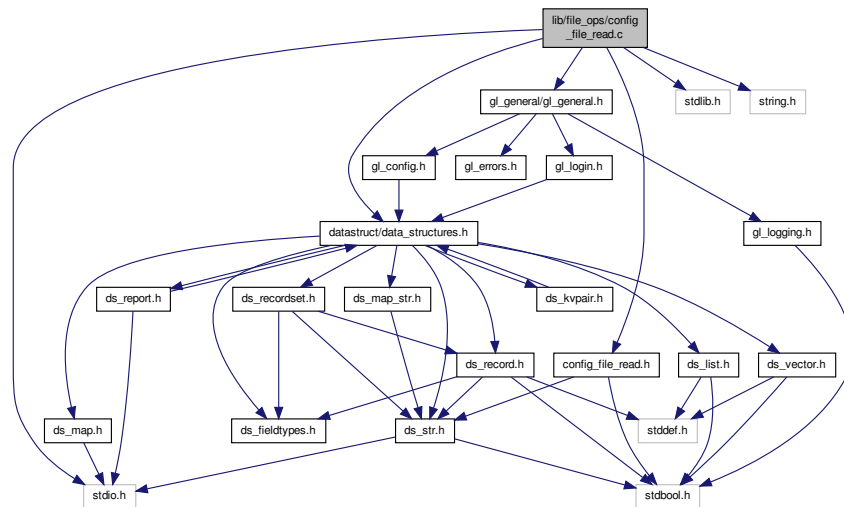
The size of the vector.

5.89 lib/file_ops/config_file_read.c File Reference

Implementation of configuration file reading functionality.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "gl_general/gl_general.h"
#include "datastruct/data_structures.h"
#include "config_file_read.h"
```

Include dependency graph for config_file_read.c:



Macros

- `#define MAX_BUFFER_SIZE 1024`
- `#define CONFIG_MAP_SIZE 100`

Functions

- `bool config_init (void)`
Initializes configuration data.
- `int config_file_read (const char *filename)`
Reads a configuration file and stores the key-value pairs.
- `ds_str config_value_get (ds_str key)`
Returns the value associated with a key.
- `ds_str config_value_get_cstr (const char *key)`
Returns the value associated with a C-style string key.
- `void config_value_set (ds_str key, ds_str value)`
Sets a key-value in the configuration structure.
- `void config_free (void)`
Frees the resources used by this module.

5.89.1 Detailed Description

Implementation of configuration file reading functionality. This module reads configuration files in the format "key = value" and makes those values available. Leading and trailing whitespace is removed for both the key and the value. Blank lines and lines starting with a '#' are ignored in the configuration file.

Author

Paul Griffiths

Copyright

Copyright 2013 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.89.2 Macro Definition Documentation

5.89.2.1 `#define CONFIG_MAP_SIZE 100`

Size to use for the hash map to contain the key-value pairs

5.89.2.2 `#define MAX_BUFFER_SIZE 1024`

Maximum size of buffers

5.89.3 Function Documentation

5.89.3.1 `int config_file_read (const char * filename)`

Reads a configuration file and stores the key-value pairs.

Parameters

<i>filename</i>	The name of the configuration file.
-----------------	-------------------------------------

Returns

`CONFIG_FILE_OK` on success, `CONFIG_FILE_NO_FILE` if the specified file could not be opened for reading, `CONFIG_FILE_MALFORMED_FILE` if the configuration file was improperly formed.

5.89.3.2 `void config_free (void)`

Frees the resources used by this module.

The user should make copies of any required keys or values prior to calling this function. This function need not be called if `config_file_read()` returned an error.

5.89.3.3 `bool config_init (void)`

Initializes configuration data.

Returns

`true` on success, `false` on failure.

5.89.3.4 `ds_str config_value_get (ds_str key)`

Returns the value associated with a key.

Parameters

<i>key</i>	The specified key.
------------	--------------------

Returns

A pointer to the associated value, or `NULL` if the key was not present in the configuration file. The caller should not modify the string to which the pointer points.

5.89.3.5 `ds_str config_value_get_cstr (const char * key)`

Returns the value associated with a C-style string key.

Parameters

<i>key</i>	The specified key.
------------	--------------------

Returns

A pointer to the associated value, or `NULL` if the key was not present in the configuration file. The caller should not modify the string to which the pointer points.

5.89.3.6 `void config_value_set (ds_str key, ds_str value)`

Sets a key-value in the configuration structure.

Parameters

<i>key</i>	The key.
<i>value</i>	The value.

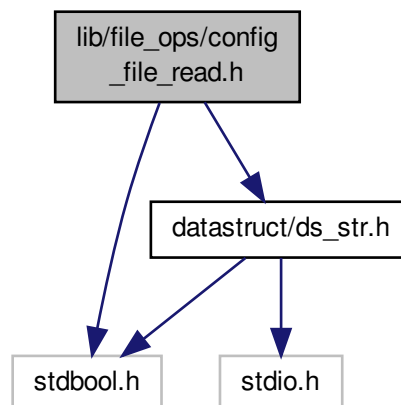
5.90 `lib/file_ops/config_file_read.h` File Reference

Interface to configuration file reading functionality.

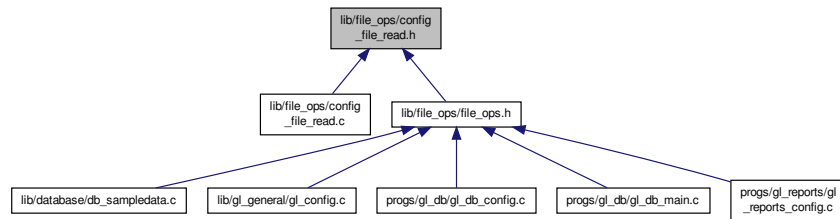
```
#include <stdbool.h>
```

```
#include "datastruct/ds_str.h"
```

Include dependency graph for `config_file_read.h`:



This graph shows which files directly or indirectly include this file:



Macros

- `#define CONFIG_FILE_OK 0`
- `#define CONFIG_FILE_NO_FILE 1`
- `#define CONFIG_FILE_MALFORMED_FILE 2`

Functions

- `bool config_init (void)`
Initializes configuration data.
- `int config_file_read (const char *filename)`
Reads a configuration file and stores the key-value pairs.
- `void config_free (void)`
Frees the resources used by this module.
- `ds_str config_value_get (ds_str key)`
Returns the value associated with a key.
- `ds_str config_value_get_cstr (const char *key)`
Returns the value associated with a C-style string key.
- `void config_value_set (ds_str key, ds_str value)`
Sets a key-value in the configuration structure.

5.90.1 Detailed Description

Interface to configuration file reading functionality. This module reads configuration files in the format "key = value" and makes those values available. Leading and trailing whitespace is removed for both the key and the value. Blank lines and lines starting with a '#' are ignored in the configuration file.

Author

Paul Griffiths

Copyright

Copyright 2013 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.90.2 Macro Definition Documentation

5.90.2.1 `#define CONFIG_FILE_MALFORMED_FILE 2`

Return status when configuration file is improperly formed

5.90.2.2 `#define CONFIG_FILE_NO_FILE 1`

Return status when unable to open file for reading

5.90.2.3 `#define CONFIG_FILE_OK 0`

Return status for success

5.90.3 Function Documentation

5.90.3.1 `int config_file_read (const char * filename)`

Reads a configuration file and stores the key-value pairs.

Parameters

<i>filename</i>	The name of the configuration file.
-----------------	-------------------------------------

Returns

`CONFIG_FILE_OK` on success, `CONFIG_FILE_NO_FILE` if the specified file could not be opened for reading, `CONFIG_FILE_MALFORMED_FILE` if the configuration file was improperly formed.

5.90.3.2 `void config_free (void)`

Frees the resources used by this module.

The user should make copies of any required keys or values prior to calling this function. This function need not be called if `config_file_read()` returned an error.

5.90.3.3 `bool config_init (void)`

Initializes configuration data.

Returns

`true` on success, `false` on failure.

5.90.3.4 `ds_str config_value_get (ds_str key)`

Returns the value associated with a key.

Parameters

<i>key</i>	The specified key.
------------	--------------------

Returns

A pointer to the associated value, or `NULL` if the key was not present in the configuration file. The caller should not modify the string to which the pointer points.

5.90.3.5 `ds_str config_value_get_cstr (const char * key)`

Returns the value associated with a C-style string key.

Parameters

<i>key</i>	The specified key.
------------	--------------------

Returns

A pointer to the associated value, or `NULL` if the key was not present in the configuration file. The caller should not modify the string to which the pointer points.

5.90.3.6 `void config_value_set (ds_str key, ds_str value)`

Sets a key-value in the configuration structure.

Parameters

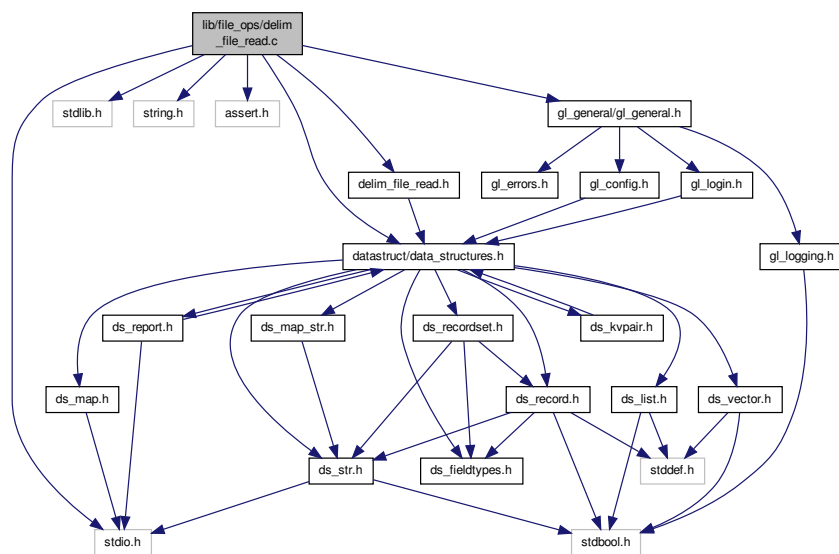
<i>key</i>	The key.
<i>value</i>	The value.

5.91 lib/file_ops/delim_file_read.c File Reference

Implementation of delimited file reading functionality.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <assert.h>
#include "gl_general/gl_general.h"
#include "datastruct/data_structures.h"
#include "delim_file_read.h"
```

Include dependency graph for `delim_file_read.c`:



Macros

- `#define MAX_LINE_SIZE 1024`

Functions

- `ds_recordset delim_file_read` (const char *filename, const char delim)
Constructs a `ds_recordset` from a delimited file.

5.91.1 Detailed Description

Implementation of delimited file reading functionality.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.91.2 Macro Definition Documentation

5.91.2.1 `#define MAX_LINE_SIZE 1024`

Maximum size of buffers

5.91.3 Function Documentation

5.91.3.1 `ds_recordset delim_file_read (const char * filename, const char delim)`

Constructs a `ds_recordset` from a delimited file.

Parameters

<i>filename</i>	The name of the delimited file.
<i>delim</i>	The delimiting character.

Returns

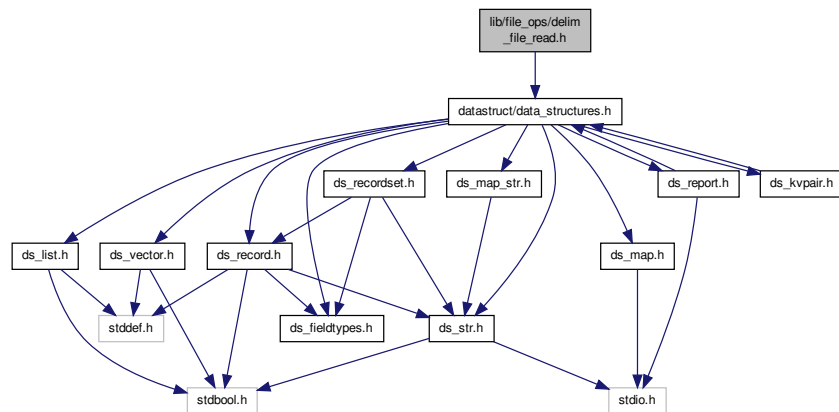
The [ds_recordset](#), or NULL on failure.

5.92 lib/file_ops/delim_file_read.h File Reference

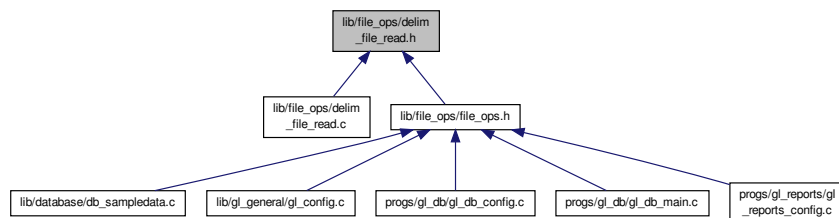
Interface to delimited file reading functionality.

```
#include "datastruct/data_structures.h"
```

Include dependency graph for `delim_file_read.h`:



This graph shows which files directly or indirectly include this file:



Functions

- [ds_recordset delim_file_read](#) (const char *filename, const char delim)
Constructs a [ds_recordset](#) from a delimited file.

5.92.1 Detailed Description

Interface to delimited file reading functionality.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.92.2 Function Documentation

5.92.2.1 `ds_recordset delim_file_read (const char * filename, const char delim)`

Constructs a `ds_recordset` from a delimited file.

Parameters

<i>filename</i>	The name of the delimited file.
<i>delim</i>	The delimiting character.

Returns

The `ds_recordset`, or `NULL` on failure.

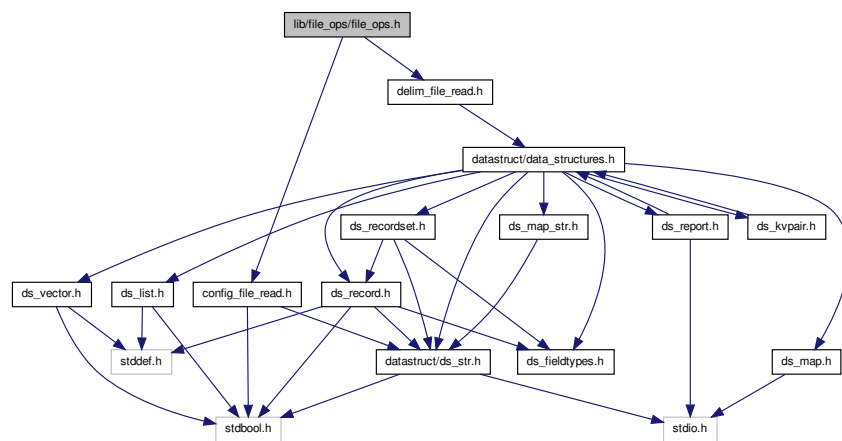
5.93 `lib/file_ops/file_ops.h` File Reference

User interface to file operations functionality.

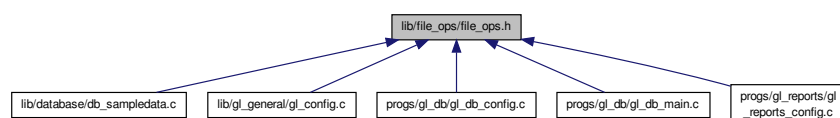
```
#include "config_file_read.h"
```

```
#include "delim_file_read.h"
```

Include dependency graph for `file_ops.h`:



This graph shows which files directly or indirectly include this file:



5.93.1 Detailed Description

User interface to file operations functionality.

Author

Paul Griffiths

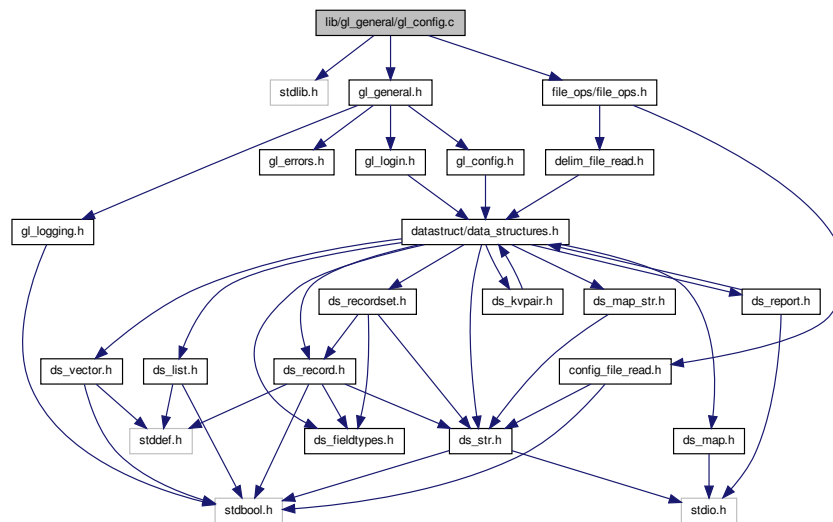
Copyright

Copyright 2013 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.94 lib/gl_general/gl_config.c File Reference

Implementation of configuration functionality.

```
#include <stdlib.h>
#include "gl_general.h"
#include "file_ops/file_ops.h"
Include dependency graph for gl_config.c:
```



Functions

- struct [params](#) * [params_init](#) (void)
Initializes a parameters structure.
- void [params_free](#) (struct [params](#) *[params](#))
Frees a parameter structure.
- bool [get_configuration](#) (struct [params](#) *[params](#), const char *[conf_file](#))
Gets parameters from a configuration file.

5.94.1 Detailed Description

Implementation of configuration functionality.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.94.2 Function Documentation**5.94.2.1 `bool get_configuration (struct params * params, const char * conf_file)`**

Gets parameters from a configuration file.

Parameters

<i>params</i>	A pointer to a parameters structure to populate.
<i>conf_file</i>	The filename of the configuration file.

Returns

`true` on success, `false` otherwise.

5.94.2.2 `void params_free (struct params * params)`

Frees a parameter structure.

Parameters

<i>params</i>	A pointer to the structure to free.
---------------	-------------------------------------

5.94.2.3 `struct params* params_init (void)` [read]

Initializes a parameters structure.

Returns

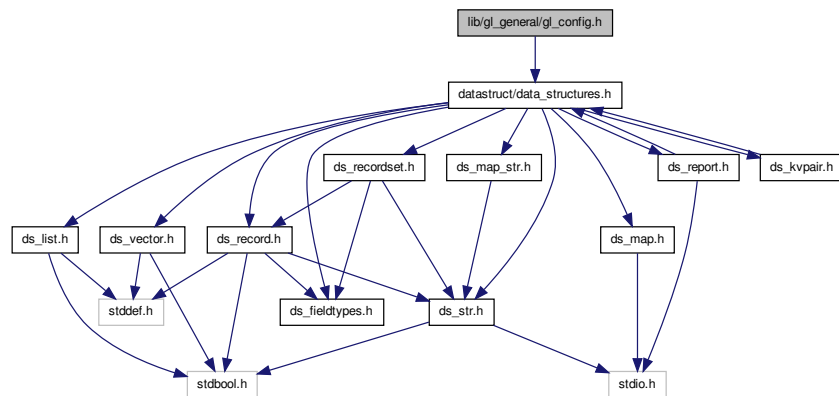
An initialized parameters structure.

5.95 lib/gl_general/gl_config.h File Reference

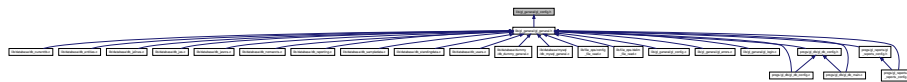
Interface to configuration functionality.

```
#include "datastruct/data_structures.h"
```

Include dependency graph for gl_config.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct [params](#)

Functions

- struct [params](#) * [params_init](#) (void)
Initializes a parameters structure.
- void [params_free](#) (struct [params](#) *[params](#))
Frees a parameter structure.
- bool [get_configuration](#) (struct [params](#) *[params](#), const char *[conf_file](#))
Gets parameters from a configuration file.

5.95.1 Detailed Description

Interface to configuration functionality.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.95.2 Function Documentation**5.95.2.1 `bool get_configuration (struct params * params, const char * conf_file)`**

Gets parameters from a configuration file.

Parameters

<i>params</i>	A pointer to a parameters structure to populate.
<i>conf_file</i>	The filename of the configuration file.

Returns

`true` on success, `false` otherwise.

5.95.2.2 `void params_free (struct params * params)`

Frees a parameter structure.

Parameters

<i>params</i>	A pointer to the structure to free.
---------------	-------------------------------------

5.95.2.3 `struct params* params_init (void)` [read]

Initializes a parameters structure.

Returns

An initialized parameters structure.

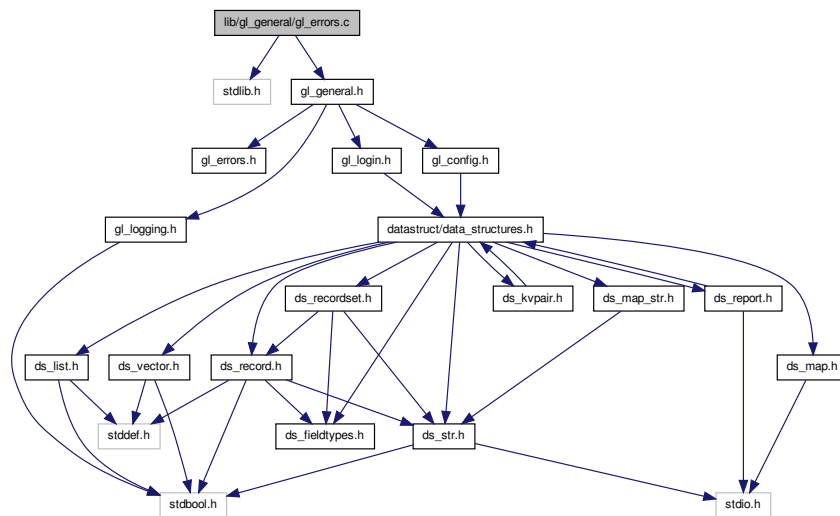
5.96 lib/gl_general/gl_errors.c File Reference

Implementation of error functionality.

```
#include <stdlib.h>
```

```
#include "gl_general.h"
```

Include dependency graph for gl_errors.c:



Functions

- void [gl_error_quit](#) (const char *msg)
Logs an error message and quits program.

5.96.1 Detailed Description

Implementation of error functionality.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.96.2 Function Documentation

5.96.2.1 void gl_error_quit (const char * msg)

Logs an error message and quits program.

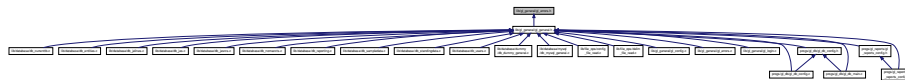
Parameters

<i>msg</i>	The error message to log.
------------	---------------------------

5.97 lib/gl_general/gl_errors.h File Reference

Interface to error functionality.

This graph shows which files directly or indirectly include this file:



Functions

- void [gl_error_quit](#) (const char *msg)
Logs an error message and quits program.

5.97.1 Detailed Description

Interface to error functionality.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.97.2 Function Documentation

5.97.2.1 void [gl_error_quit](#) (const char * *msg*)

Logs an error message and quits program.

Parameters

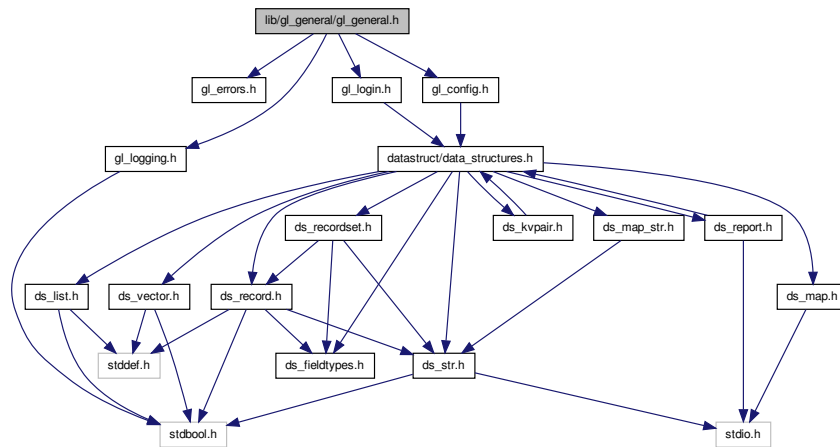
<i>msg</i>	The error message to log.
------------	---------------------------

5.98 lib/gl_general/gl_general.h File Reference

User interface to logging and error functionality.

```
#include "gl_errors.h"
#include "gl_logging.h"
#include "gl_login.h"
#include "gl_config.h"
```

Include dependency graph for gl_general.h:



This graph shows which files directly or indirectly include this file:



5.98.1 Detailed Description

User interface to logging and error functionality.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.99 lib/gl_general/gl_logging.c File Reference

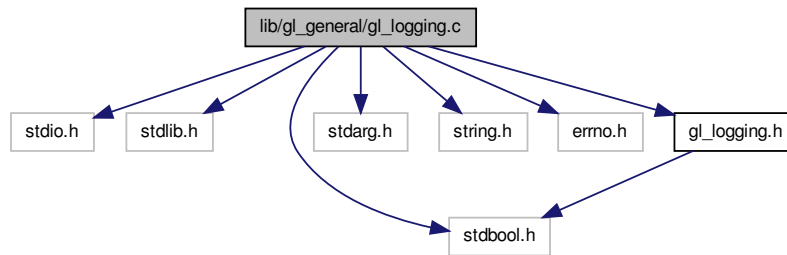
Implementation of logging functionality.

```

#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#include <stdarg.h>
#include <string.h>
#include <errno.h>
#include "gl_logging.h"

```

Include dependency graph for `gl_logging.c`:



Functions

- void `gl_set_logging` (const bool status)
Turns logging on or off.
- void `gl_log_msg` (const char *format,...)
Logs a message to the log file.

5.99.1 Detailed Description

Implementation of logging functionality. Implementation of logging functionality. Enables debugging and other system messages to be recorded to a log file.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.99.2 Function Documentation

5.99.2.1 void `gl_log_msg` (const char * *format*, ...)

Logs a message to the log file.

Logs a message to the log file.

Parameters

<i>format</i>	Format string, in same format as <code>printf()</code> .
...	Variable arguments as specified by format string.

5.99.2.2 void `gl_set_logging` (const bool *status*)

Turns logging on or off.

Turns logging on or off.

Parameters

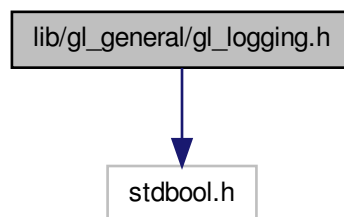
<i>status</i>	true to turn logging on, false to turn logging off.
---------------	---

5.100 lib/gl_general/gl_logging.h File Reference

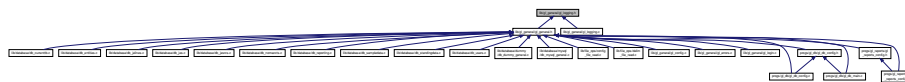
Interface to logging functionality.

```
#include <stdbool.h>
```

Include dependency graph for gl_logging.h:



This graph shows which files directly or indirectly include this file:



Functions

- void [gl_set_logging](#) (const bool status)
Turns logging on or off.
- void [gl_log_msg](#) (const char *format,...)
Logs a message to the log file.

5.100.1 Detailed Description

Interface to logging functionality. Interface to logging functionality. Enables debugging and other system messages to be recorded to a log file.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.100.2 Function Documentation

5.100.2.1 void gl_log_msg (const char * *format*, ...)

Logs a message to the log file.

Logs a message to the log file.

Parameters

<i>format</i>	Format string, in same format as <code>printf()</code> .
...	Variable arguments as specified by format string.

5.100.2.2 void gl_set_logging (const bool *status*)

Turns logging on or off.

Turns logging on or off.

Parameters

<i>status</i>	true to turn logging on, false to turn logging off.
---------------	---

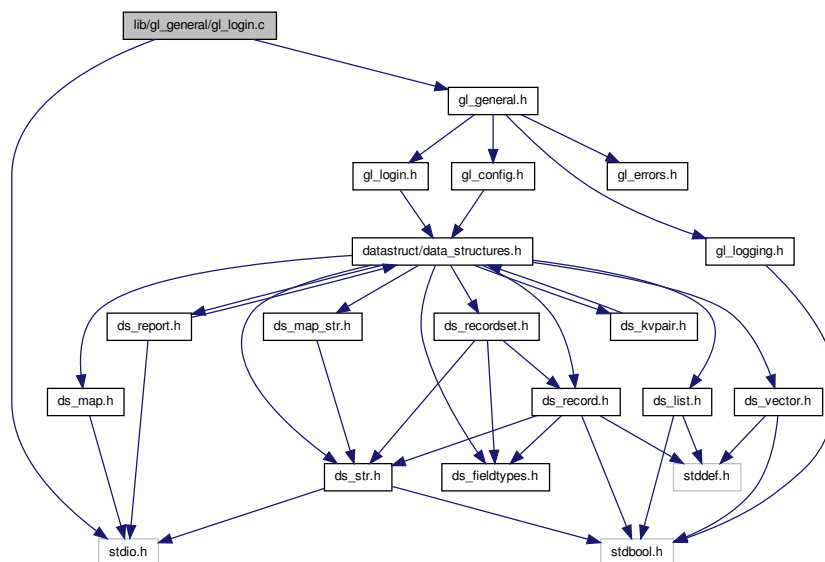
5.101 lib/gl_general/gl_login.c File Reference

Implementation of login functionality.

```
#include <stdio.h>
```

```
#include "gl_general.h"
```

Include dependency graph for `gl_login.c`:



Functions

- [ds_str login](#) (void)

Gets a password from the user.

5.101.1 Detailed Description

Implementation of login functionality.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.101.2 Function Documentation

5.101.2.1 ds_str login (void)

Gets a password from the user.

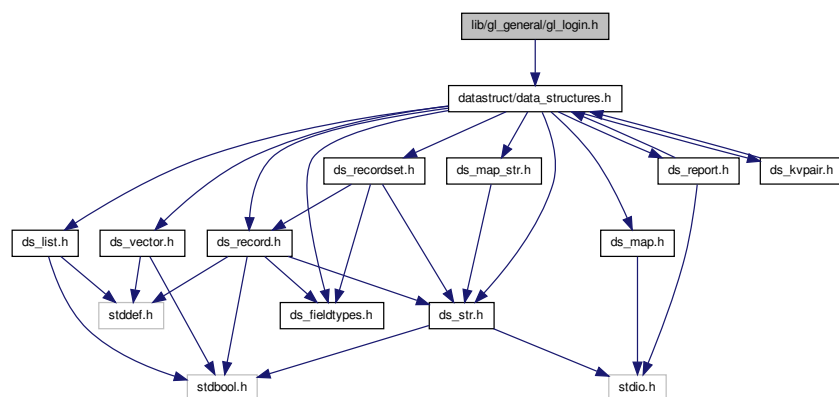
Returns

The password, or `NULL` on failure.

5.102 lib/gl_general/gl_login.h File Reference

Interface to login functionality.

```
#include "datastruct/data_structures.h"
Include dependency graph for gl_login.h:
```



This graph shows which files directly or indirectly include this file:



Functions

- [ds_str login](#) (void)

Gets a password from the user.

5.102.1 Detailed Description

Interface to login functionality.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.102.2 Function Documentation

5.102.2.1 ds_str login (void)

Gets a password from the user.

Returns

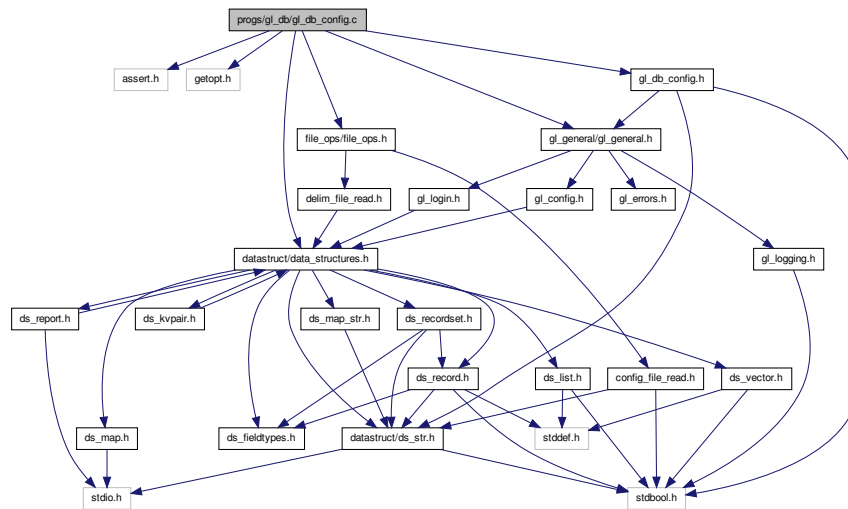
The password, or `NULL` on failure.

5.103 progs/gl_db/gl_db_config.c File Reference

Implementation of GL DB program configuration functionality.

```
#include <assert.h>
#include <getopt.h>
#include "gl_db_config.h"
#include "file_ops/file_ops.h"
#include "datastruct/data_structures.h"
#include "gl_general/gl_general.h"
```


Include dependency graph for gl_db_config.c:



Macros

- `#define _XOPEN_SOURCE 500`

Functions

- `bool get_cmdline_options (int argc, char **argv, struct params *params)`

Gets parameters from the command line.

5.103.1 Detailed Description

Implementation of GL DB program configuration functionality. Gets program configuration options from the command line and/or a configuration file.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.103.2 Macro Definition Documentation

5.103.2.1 `#define _XOPEN_SOURCE 500`

UNIX feature test macro

5.103.3 Function Documentation

5.103.3.1 `bool get_cmdline_options (int argc, char ** argv, struct params * params)`

Gets parameters from the command line.

Parameters

<i>argc</i>	<code>argc</code> as passed to <code>main()</code> .
<i>argv</i>	<code>argv</code> as passed to <code>main()</code> .
<i>params</i>	A pointer to a parameters structure to populate.

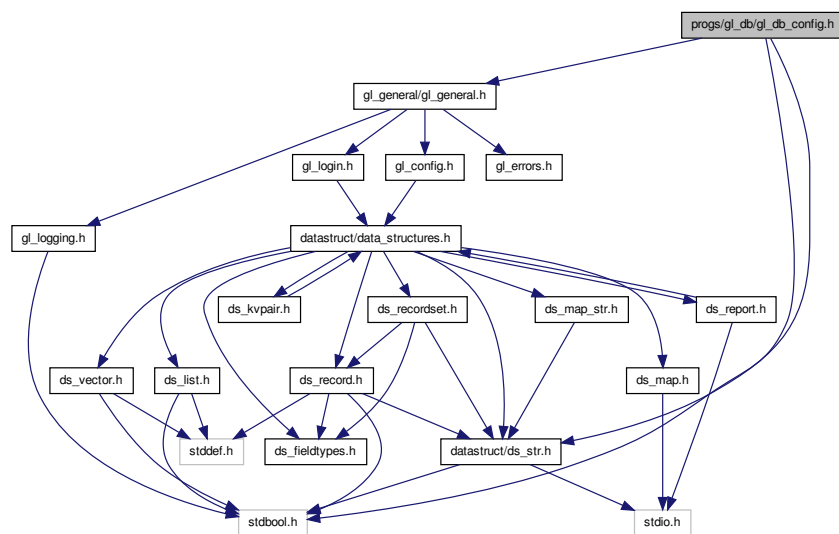
Returns

`false` if an unrecognized command line option was specified, `true` otherwise.

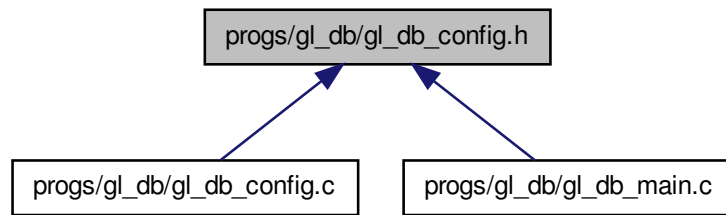
5.104 `progs/gl_db/gl_db_config.h` File Reference

Interface to GL DB program configuration functionality.

```
#include <stdbool.h>
#include "datastruct/ds_str.h"
#include "gl_general/gl_general.h"
Include dependency graph for gl_db_config.h:
```



This graph shows which files directly or indirectly include this file:



Functions

- bool `get_cmdline_options` (int argc, char **argv, struct params *params)

Gets parameters from the command line.

5.104.1 Detailed Description

Interface to GL DB program configuration functionality. Gets program configuration options from the command line and/or a configuration file.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.104.2 Function Documentation

5.104.2.1 bool `get_cmdline_options` (int argc, char ** argv, struct params * params)

Gets parameters from the command line.

Parameters

<i>argc</i>	argc as passed to <code>main()</code> .
<i>argv</i>	argv as passed to <code>main()</code> .
<i>params</i>	A pointer to a parameters structure to populate.

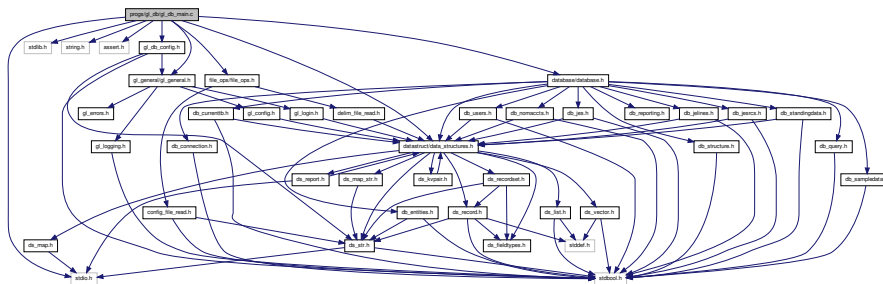
Returns

`false` if an unrecognized command line option was specified, `true` otherwise.

5.105 progs/gl_db/gl_db_main.c File Reference

Main function for GL database program.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <assert.h>
#include "gl_general/gl_general.h"
#include "database/database.h"
#include "gl_db_config.h"
#include "datastruct/data_structures.h"
#include "file_ops/file_ops.h"
Include dependency graph for gl_db_main.c:
```



Functions

- void [print_usage_message](#) (const char *progname)
Prints a program usage message.
- void [print_version_message](#) (const char *progname)
Prints a program version message.
- void [print_help_message](#) (const char *progname)
Prints a program help message.
- int [main](#) (int argc, char **argv)
Main function.

5.105.1 Detailed Description

Main function for GL database program. Main function for GL database program.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.105.2 Function Documentation

5.105.2.1 `int main (int argc, char ** argv)`

Main function.

Main function.

Returns

Exit status.

5.105.2.2 `void print_help_message (const char * progrname)`

Prints a program help message.

Parameters

<i>progrname</i>	The program name.
------------------	-------------------

5.105.2.3 `void print_usage_message (const char * progrname)`

Prints a program usage message.

Parameters

<i>progrname</i>	The program name.
------------------	-------------------

5.105.2.4 `void print_version_message (const char * progrname)`

Prints a program version message.

Parameters

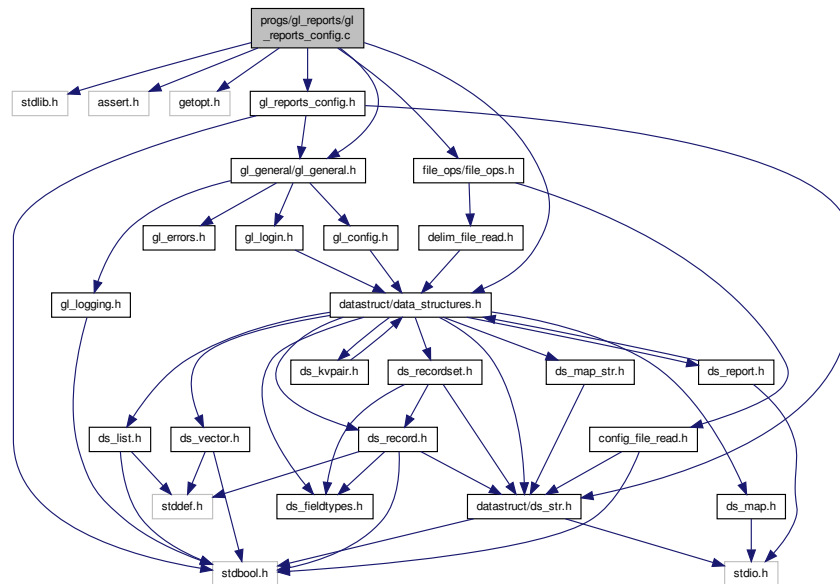
<i>progrname</i>	The program name.
------------------	-------------------

5.106 progs/gl_reports/gl_reports_config.c File Reference

Implementation of GL reports program configuration functionality.

```
#include <stdlib.h>
#include <assert.h>
#include <getopt.h>
#include "gl_reports_config.h"
#include "file_ops/file_ops.h"
#include "datastruct/data_structures.h"
#include "gl_general/gl_general.h"
```

Include dependency graph for `gl_reports_config.c`:



Macros

- `#define _XOPEN_SOURCE 500`

Functions

- `bool get_cmdline_options (int argc, char **argv, struct params *params)`
Gets parameters from the command line.

5.106.1 Detailed Description

Implementation of GL reports program configuration functionality. Gets program configuration options from the command line and/or a configuration file.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.106.2 Macro Definition Documentation

5.106.2.1 `#define _XOPEN_SOURCE 500`

UNIX feature test macro

5.106.3 Function Documentation

5.106.3.1 bool get_cmdline_options (int argc, char ** argv, struct params * params)

Gets parameters from the command line.

Parameters

<i>argc</i>	argc as passed to <code>main()</code> .
<i>argv</i>	argv as passed to <code>main()</code> .
<i>params</i>	A pointer to a parameters structure to populate.

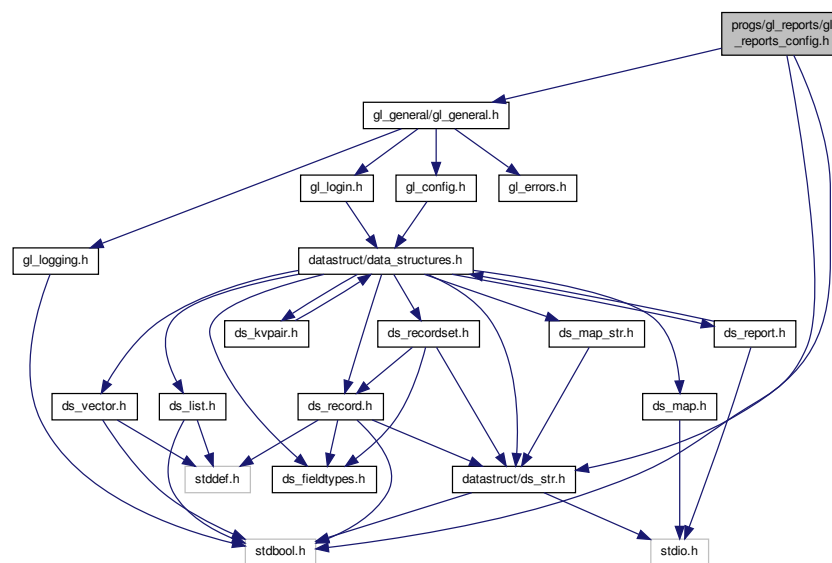
Returns

`false` if an unrecognized command line option was specified, `true` otherwise.

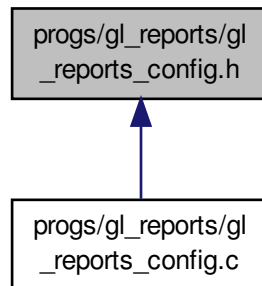
5.107 progs/gl_reports/gl_reports_config.h File Reference

Interface to GL reports program configuration functionality.

```
#include <stdbool.h>
#include "datastruct/ds_str.h"
#include "gl_general/gl_general.h"
Include dependency graph for gl_reports_config.h:
```



This graph shows which files directly or indirectly include this file:



Functions

- bool `get_cmdline_options` (int argc, char **argv, struct `params` *params)
Gets parameters from the command line.

5.107.1 Detailed Description

Interface to GL reports program configuration functionality. Gets program configuration options from the command line and/or a configuration file.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

5.107.2 Function Documentation

5.107.2.1 bool `get_cmdline_options` (int argc, char ** argv, struct `params` * params)

Gets parameters from the command line.

Parameters

<i>argc</i>	argc as passed to <code>main()</code> .
<i>argv</i>	argv as passed to <code>main()</code> .
<i>params</i>	A pointer to a parameters structure to populate.

Returns

`false` if an unrecognized command line option was specified, `true` otherwise.

Index

- `_XOPEN_SOURCE`
 - `db_dummy_general.c`, 69
 - `ds_report.c`, 133
 - `gl_db_config.c`, 185
 - `gl_reports_config.c`, 190
- `CONFIG_FILE_OK`
 - `config_file_read.h`, 168
- `CONFIG_MAP_SIZE`
 - `config_file_read.c`, 165
- `capacity`
 - `ds_str`, 19
- `config_file_read`
 - `config_file_read.c`, 165
 - `config_file_read.h`, 168
- `config_file_read.c`
 - `CONFIG_MAP_SIZE`, 165
 - `config_file_read`, 165
 - `config_free`, 165
 - `config_init`, 165
 - `config_value_get`, 165
 - `config_value_get_cstr`, 165
 - `config_value_set`, 166
 - `MAX_BUFFER_SIZE`, 165
- `config_file_read.h`
 - `CONFIG_FILE_OK`, 168
 - `config_file_read`, 168
 - `config_free`, 168
 - `config_init`, 168
 - `config_value_get`, 168
 - `config_value_get_cstr`, 168
 - `config_value_set`, 169
- `config_free`
 - `config_file_read.c`, 165
 - `config_file_read.h`, 168
- `config_init`
 - `config_file_read.c`, 165
 - `config_file_read.h`, 168
- `config_value_get`
 - `config_file_read.c`, 165
 - `config_file_read.h`, 168
- `config_value_get_cstr`
 - `config_file_read.c`, 165
 - `config_file_read.h`, 168
- `config_value_set`
 - `config_file_read.c`, 166
 - `config_file_read.h`, 169
- `conn_mss`
 - `db_mysql_general.c`, 89
- `created_time`
 - `ds_report`, 19
- `current`
 - `ds_list`, 12
 - `ds_vector`, 20
- `DS_FIELD_BOOLEAN`
 - `ds_fieldtypes.h`, 95
- `DS_FIELD_DOUBLE`
 - `ds_fieldtypes.h`, 95
- `DS_FIELD_INT`
 - `ds_fieldtypes.h`, 95
- `DS_FIELD_STRING`
 - `ds_fieldtypes.h`, 95
- `data`
 - `ds_list_element`, 13
 - `ds_str`, 19
 - `ds_vector`, 20
- `data_destructor`
 - `ds_list`, 12
 - `ds_vector`, 20
- `database`
 - `params`, 22
- `db_all_jes_number_report_sql`
 - `db_mysql_all_jes_number_report_sql.c`, 72
 - `db_sql.h`, 53
- `db_all_jes_report`
 - `db_jes.c`, 37
 - `db_jes.h`, 39
- `db_all_jes_report_sql`
 - `db_mysql_all_jes_report_sql.c`, 72
 - `db_sql.h`, 53
- `db_check_total_entity_report_sql`
 - `db_mysql_check_total_entity_report_sql.c`, 73
 - `db_sql.h`, 53
- `db_check_total_report`
 - `db_currenttb.c`, 26
 - `db_currenttb.h`, 28
- `db_check_total_report_sql`
 - `db_mysql_check_total_report_sql.c`, 73
 - `db_sql.h`, 53
- `db_connect`
 - `db_connection.h`, 25
 - `db_dummy_general.c`, 69
 - `db_mysql_general.c`, 88
- `db_connection.h`
 - `db_connect`, 25
- `db_create_all_jes_view`
 - `db_jes.c`, 37
 - `db_jes.h`, 39
- `db_create_all_jes_view_sql`

- db_mysql_create_all_jes_view_sql.c, 74
- db_sql.h, 53
- db_create_check_total_view
 - db_currenttb.c, 26
 - db_currenttb.h, 28
- db_create_check_total_view_sql
 - db_mysql_create_check_total_view_sql.c, 75
 - db_sql.h, 53
- db_create_current_trial_balance_view
 - db_currenttb.c, 26
 - db_currenttb.h, 28
- db_create_current_trial_balance_view_sql
 - db_sql.h, 53
- db_create_database_structure
 - db_structure.c, 61
 - db_structure.h, 62
- db_create_entities_table
 - db_entities.c, 30
 - db_entities.h, 32
- db_create_entities_table_sql
 - db_dummy_create_entities_table_sql.c, 66
 - db_mysql_create_entities_table_sql.c, 76
 - db_sql.h, 54
- db_create_jelines_table
 - db_jelines.c, 34
 - db_jelines.h, 36
- db_create_jelines_table_sql
 - db_mysql_create_jelines_table_sql.c, 76
 - db_sql.h, 54
- db_create_jes_table
 - db_jes.c, 37
 - db_jes.h, 40
- db_create_jes_table_sql
 - db_mysql_create_jes_table_sql.c, 77
 - db_sql.h, 54
- db_create_jesrcs_table
 - db_jesrcs.c, 41
 - db_jesrcs.h, 43
- db_create_jesrcs_table_sql
 - db_mysql_create_jesrcs_table_sql.c, 78
 - db_sql.h, 54
- db_create_nomaccts_table
 - db_nomaccts.c, 44
 - db_nomaccts.h, 45
- db_create_nomaccts_table_sql
 - db_mysql_create_nomaccts_table_sql.c, 78
 - db_sql.h, 54
- db_create_recordset_from_query
 - db_dummy_general.c, 69
 - db_mysql_general.c, 89
 - db_reporting.h, 49
- db_create_report_from_query
 - db_reporting.c, 48
 - db_reporting.h, 49
- db_create_standingdata_table
 - db_standingdata.c, 58
 - db_standingdata.h, 60
- db_create_standingdata_table_sql
 - db_mysql_create_standingdata_table_sql.c, 79
 - db_sql.h, 54
- db_create_users_table
 - db_users.c, 64
 - db_users.h, 65
- db_create_users_table_sql
 - db_dummy_create_users_table_sql.c, 67
 - db_mysql_create_users_table_sql.c, 79
 - db_sql.h, 54
- db_current_trial_balance_entity_report_sql
 - db_sql.h, 55
- db_current_trial_balance_report
 - db_currenttb.c, 26
 - db_currenttb.h, 28
- db_current_trial_balance_report_sql
 - db_mysql_current_trial_balance_report_sql.c, 81
 - db_sql.h, 55
- db_currenttb.c
 - db_check_total_report, 26
 - db_create_check_total_view, 26
 - db_create_current_trial_balance_view, 26
 - db_current_trial_balance_report, 26
 - db_drop_check_total_view, 26
 - db_drop_current_trial_balance_view, 27
- db_currenttb.h
 - db_check_total_report, 28
 - db_create_check_total_view, 28
 - db_create_current_trial_balance_view, 28
 - db_current_trial_balance_report, 28
 - db_drop_check_total_view, 28
 - db_drop_current_trial_balance_view, 29
- db_delete_database_structure
 - db_structure.c, 61
 - db_structure.h, 63
- db_drop_all_jes_view
 - db_jes.c, 38
 - db_jes.h, 40
- db_drop_all_jes_view_sql
 - db_mysql_drop_all_jes_view_sql.c, 81
 - db_sql.h, 55
- db_drop_check_total_view
 - db_currenttb.c, 26
 - db_currenttb.h, 28
- db_drop_check_total_view_sql
 - db_mysql_drop_check_total_view_sql.c, 82
 - db_sql.h, 55
- db_drop_current_trial_balance_view
 - db_currenttb.c, 27
 - db_currenttb.h, 29
- db_drop_current_trial_balance_view_sql
 - db_sql.h, 55
- db_drop_entities_table
 - db_entities.c, 30
 - db_entities.h, 32
- db_drop_entities_table_sql
 - db_dummy_drop_entities_table_sql.c, 67
 - db_mysql_drop_entities_table_sql.c, 83
 - db_sql.h, 55

- db_drop_jelines_table
 - db_jelines.c, [34](#)
 - db_jelines.h, [36](#)
- db_drop_jelines_table_sql
 - db_mysql_drop_jelines_table_sql.c, [84](#)
 - db_sql.h, [55](#)
- db_drop_jes_table
 - db_jes.c, [38](#)
 - db_jes.h, [40](#)
- db_drop_jes_table_sql
 - db_mysql_drop_jes_table_sql.c, [84](#)
 - db_sql.h, [56](#)
- db_drop_jesrcs_table
 - db_jesrcs.c, [41](#)
 - db_jesrcs.h, [43](#)
- db_drop_jesrcs_table_sql
 - db_mysql_drop_jesrcs_table_sql.c, [85](#)
 - db_sql.h, [56](#)
- db_drop_nomaccts_table
 - db_nomaccts.c, [44](#)
 - db_nomaccts.h, [46](#)
- db_drop_nomaccts_table_sql
 - db_mysql_drop_nomaccts_table_sql.c, [85](#)
 - db_sql.h, [56](#)
- db_drop_standingdata_table
 - db_standingdata.c, [58](#)
 - db_standingdata.h, [60](#)
- db_drop_standingdata_table_sql
 - db_mysql_drop_standingdata_table_sql.c, [86](#)
 - db_sql.h, [56](#)
- db_drop_users_table
 - db_users.c, [64](#)
 - db_users.h, [65](#)
- db_drop_users_table_sql
 - db_dummy_drop_users_table_sql.c, [68](#)
 - db_mysql_drop_users_table_sql.c, [87](#)
 - db_sql.h, [56](#)
- db_dummy_create_entities_table_sql.c
 - db_create_entities_table_sql, [66](#)
- db_dummy_create_users_table_sql.c
 - db_create_users_table_sql, [67](#)
- db_dummy_drop_entities_table_sql.c
 - db_drop_entities_table_sql, [67](#)
- db_dummy_drop_users_table_sql.c
 - db_drop_users_table_sql, [68](#)
- db_dummy_general.c
 - _XOPEN_SOURCE, [69](#)
 - db_connect, [69](#)
 - db_create_recordset_from_query, [69](#)
 - db_execute_query, [70](#)
- db_dummy_list_entities_report_sql.c
 - db_list_entities_report_sql, [70](#)
- db_dummy_list_users_report_sql.c
 - db_list_users_report_sql, [71](#)
- db_entities.c
 - db_create_entities_table, [30](#)
 - db_drop_entities_table, [30](#)
 - db_get_entity_name_from_id, [30](#)
- db_list_entities_report, [30](#)
- db_entities.h
 - db_create_entities_table, [32](#)
 - db_drop_entities_table, [32](#)
 - db_get_entity_name_from_id, [32](#)
 - db_list_entities_report, [32](#)
- db_execute_query
 - db_dummy_general.c, [70](#)
 - db_mysql_general.c, [89](#)
 - db_query.h, [47](#)
- db_get_entity_name_from_id
 - db_entities.c, [30](#)
 - db_entities.h, [32](#)
- db_get_entity_name_from_id_sql
 - db_mysql_entity_name_from_id_sql.c, [87](#)
 - db_sql.h, [56](#)
- db_jelines.c
 - db_create_jelines_table, [34](#)
 - db_drop_jelines_table, [34](#)
 - db_list_jelines_report, [34](#)
- db_jelines.h
 - db_create_jelines_table, [36](#)
 - db_drop_jelines_table, [36](#)
 - db_list_jelines_report, [36](#)
- db_jes.c
 - db_all_jes_report, [37](#)
 - db_create_all_jes_view, [37](#)
 - db_create_jes_table, [37](#)
 - db_drop_all_jes_view, [38](#)
 - db_drop_jes_table, [38](#)
 - db_list_jes_report, [38](#)
- db_jes.h
 - db_all_jes_report, [39](#)
 - db_create_all_jes_view, [39](#)
 - db_create_jes_table, [40](#)
 - db_drop_all_jes_view, [40](#)
 - db_drop_jes_table, [40](#)
 - db_list_jes_report, [40](#)
- db_jesrcs.c
 - db_create_jesrcs_table, [41](#)
 - db_drop_jesrcs_table, [41](#)
 - db_list_jesrcs_report, [41](#)
- db_jesrcs.h
 - db_create_jesrcs_table, [43](#)
 - db_drop_jesrcs_table, [43](#)
 - db_list_jesrcs_report, [43](#)
- db_list_entities_report
 - db_entities.c, [30](#)
 - db_entities.h, [32](#)
- db_list_entities_report_sql
 - db_dummy_list_entities_report_sql.c, [70](#)
 - db_mysql_list_entities_report_sql.c, [90](#)
 - db_sql.h, [56](#)
- db_list_jelines_report
 - db_jelines.c, [34](#)
 - db_jelines.h, [36](#)
- db_list_jelines_report_sql
 - db_mysql_list_jelines_report_sql.c, [90](#)

- db_sql.h, 57
- db_list_jes_report
 - db_jes.c, 38
 - db_jes.h, 40
- db_list_jes_report_sql
 - db_mysql_list_jes_report_sql.c, 91
 - db_sql.h, 57
- db_list_jesrcs_report
 - db_jesrcs.c, 41
 - db_jesrcs.h, 43
- db_list_jesrcs_report_sql
 - db_mysql_list_jesrcs_report_sql.c, 92
 - db_sql.h, 57
- db_list_nomaccts_report
 - db_nomaccts.c, 44
 - db_nomaccts.h, 46
- db_list_nomaccts_report_sql
 - db_mysql_list_nomaccts_report_sql.c, 92
 - db_sql.h, 57
- db_list_users_report
 - db_users.c, 64
 - db_users.h, 65
- db_list_users_report_sql
 - db_dummy_list_users_report_sql.c, 71
 - db_mysql_list_users_report_sql.c, 93
 - db_sql.h, 57
- db_mysql_all_jes_report_sql.c
 - db_all_jes_report_sql, 72
- db_mysql_check_total_report_sql.c
 - db_check_total_report_sql, 73
- db_mysql_create_entities_table_sql.c
 - db_create_entities_table_sql, 76
- db_mysql_create_jelines_table_sql.c
 - db_create_jelines_table_sql, 76
- db_mysql_create_jes_table_sql.c
 - db_create_jes_table_sql, 77
- db_mysql_create_jesrcs_table_sql.c
 - db_create_jesrcs_table_sql, 78
- db_mysql_create_nomaccts_table_sql.c
 - db_create_nomaccts_table_sql, 78
- db_mysql_create_standingdata_table_sql.c
 - db_create_standingdata_table_sql, 79
- db_mysql_create_users_table_sql.c
 - db_create_users_table_sql, 79
- db_mysql_drop_entities_table_sql.c
 - db_drop_entities_table_sql, 83
- db_mysql_drop_jelines_table_sql.c
 - db_drop_jelines_table_sql, 84
- db_mysql_drop_jes_table_sql.c
 - db_drop_jes_table_sql, 84
- db_mysql_drop_jesrcs_table_sql.c
 - db_drop_jesrcs_table_sql, 85
- db_mysql_drop_nomaccts_table_sql.c
 - db_drop_nomaccts_table_sql, 85
- db_mysql_drop_standingdata_table_sql.c
 - db_drop_standingdata_table_sql, 86
- db_mysql_drop_users_table_sql.c
 - db_drop_users_table_sql, 87
- db_mysql_general.c
 - conn_mss, 89
 - db_connect, 88
 - db_create_recordset_from_query, 89
 - db_execute_query, 89
 - main_mss, 89
- db_mysql_list_entities_report_sql.c
 - db_list_entities_report_sql, 90
- db_mysql_list_jelines_report_sql.c
 - db_list_jelines_report_sql, 90
- db_mysql_list_jes_report_sql.c
 - db_list_jes_report_sql, 91
- db_mysql_list_jesrcs_report_sql.c
 - db_list_jesrcs_report_sql, 92
- db_mysql_list_nomaccts_report_sql.c
 - db_list_nomaccts_report_sql, 92
- db_mysql_list_users_report_sql.c
 - db_list_users_report_sql, 93
- db_mysql_show_standingdata_report_sql.c
 - db_show_standingdata_report_sql, 93
- db_nomaccts.c
 - db_create_nomaccts_table, 44
 - db_drop_nomaccts_table, 44
 - db_list_nomaccts_report, 44
- db_nomaccts.h
 - db_create_nomaccts_table, 45
 - db_drop_nomaccts_table, 46
 - db_list_nomaccts_report, 46
- db_query.h
 - db_execute_query, 47
- db_reporting.c
 - db_create_report_from_query, 48
- db_reporting.h
 - db_create_recordset_from_query, 49
 - db_create_report_from_query, 49
- db_show_standingdata_report
 - db_standingdata.c, 59
 - db_standingdata.h, 60
- db_show_standingdata_report_sql
 - db_mysql_show_standingdata_report_sql.c, 93
 - db_sql.h, 57
- db_sql.h
 - db_all_jes_number_report_sql, 53
 - db_all_jes_report_sql, 53
 - db_check_total_entity_report_sql, 53
 - db_check_total_report_sql, 53
 - db_create_all_jes_view_sql, 53
 - db_create_check_total_view_sql, 53
 - db_create_current_trial_balance_view_sql, 53
 - db_create_entities_table_sql, 54
 - db_create_jelines_table_sql, 54
 - db_create_jes_table_sql, 54
 - db_create_jesrcs_table_sql, 54
 - db_create_nomaccts_table_sql, 54
 - db_create_standingdata_table_sql, 54
 - db_create_users_table_sql, 54
 - db_current_trial_balance_entity_report_sql, 55
 - db_current_trial_balance_report_sql, 55

- db_drop_all_jes_view_sql, 55
- db_drop_check_total_view_sql, 55
- db_drop_current_trial_balance_view_sql, 55
- db_drop_entities_table_sql, 55
- db_drop_jelines_table_sql, 55
- db_drop_jes_table_sql, 56
- db_drop_jesrcs_table_sql, 56
- db_drop_nomaccts_table_sql, 56
- db_drop_standingdata_table_sql, 56
- db_drop_users_table_sql, 56
- db_get_entity_name_from_id_sql, 56
- db_list_entities_report_sql, 56
- db_list_jelines_report_sql, 57
- db_list_jes_report_sql, 57
- db_list_jesrcs_report_sql, 57
- db_list_nomaccts_report_sql, 57
- db_list_users_report_sql, 57
- db_show_standingdata_report_sql, 57
- db_standingdata.c
 - db_create_standingdata_table, 58
 - db_drop_standingdata_table, 58
 - db_show_standingdata_report, 59
- db_standingdata.h
 - db_create_standingdata_table, 60
 - db_drop_standingdata_table, 60
 - db_show_standingdata_report, 60
- db_structure.c
 - db_create_database_structure, 61
 - db_delete_database_structure, 61
- db_structure.h
 - db_create_database_structure, 62
 - db_delete_database_structure, 63
- db_users.c
 - db_create_users_table, 64
 - db_drop_users_table, 64
 - db_list_users_report, 64
- db_users.h
 - db_create_users_table, 65
 - db_drop_users_table, 65
 - db_list_users_report, 65
- delim_file_read
 - delim_file_read.c, 170
 - delim_file_read.h, 172
- delim_file_read.c
 - delim_file_read, 170
 - MAX_LINE_SIZE, 170
- delim_file_read.h
 - delim_file_read, 172
- ds_fieldtypes.h
 - DS_FIELD_BOOLEAN, 95
 - DS_FIELD_DOUBLE, 95
 - DS_FIELD_INT, 95
 - DS_FIELD_STRING, 95
- ds_field_types
 - ds_fieldtypes.h, 95
- ds_fieldtypes.h
 - ds_field_types, 95
- ds_kvpair, 11
 - ds_kvpair.h, 97
 - key, 11
 - value, 11
- ds_kvpair.h
 - ds_kvpair, 97
 - ds_kvpair_create, 97
 - ds_kvpair_destroy, 97
 - ds_kvpair_destructor, 97
 - ds_kvpair_get_key, 97
 - ds_kvpair_get_value, 98
- ds_kvpair_create
 - ds_kvpair.h, 97
- ds_kvpair_destroy
 - ds_kvpair.h, 97
- ds_kvpair_destructor
 - ds_kvpair.h, 97
- ds_kvpair_get_key
 - ds_kvpair.h, 97
- ds_kvpair_get_value
 - ds_kvpair.h, 98
- ds_list, 12
 - current, 12
 - data_destructor, 12
 - ds_list.h, 104
 - free_on_delete, 12
 - head, 13
 - length, 13
 - tail, 13
- ds_list.c
 - ds_list_append, 99
 - ds_list_create, 100
 - ds_list_destroy, 100
 - ds_list_destructor, 100
 - ds_list_element, 100
 - ds_list_get_next_data, 100
 - ds_list_get_prev_data, 101
 - ds_list_is_empty, 101
 - ds_list_length, 101
 - ds_list_remove_all, 101
 - ds_list_remove_tail, 102
 - ds_list_seek_end, 102
 - ds_list_seek_start, 102
- ds_list.h
 - ds_list, 104
 - ds_list_append, 104
 - ds_list_create, 104
 - ds_list_destroy, 104
 - ds_list_destructor, 104
 - ds_list_element, 105
 - ds_list_get_next_data, 105
 - ds_list_get_prev_data, 105
 - ds_list_is_empty, 105
 - ds_list_length, 106
 - ds_list_remove_all, 106
 - ds_list_remove_tail, 106
 - ds_list_seek_end, 106
 - ds_list_seek_start, 106
- ds_list_append

- ds_list.c, 99
 - ds_list.h, 104
- ds_list_create
 - ds_list.c, 100
 - ds_list.h, 104
- ds_list_destroy
 - ds_list.c, 100
 - ds_list.h, 104
- ds_list_destructor
 - ds_list.c, 100
 - ds_list.h, 104
- ds_list_element, 13
 - data, 13
 - ds_list.c, 100
 - ds_list.h, 105
 - next, 13
 - previous, 14
- ds_list_get_next_data
 - ds_list.c, 100
 - ds_list.h, 105
- ds_list_get_prev_data
 - ds_list.c, 101
 - ds_list.h, 105
- ds_list_is_empty
 - ds_list.c, 101
 - ds_list.h, 105
- ds_list_length
 - ds_list.c, 101
 - ds_list.h, 106
- ds_list_remove_all
 - ds_list.c, 101
 - ds_list.h, 106
- ds_list_remove_tail
 - ds_list.c, 102
 - ds_list.h, 106
- ds_list_seek_end
 - ds_list.c, 102
 - ds_list.h, 106
- ds_list_seek_start
 - ds_list.c, 102
 - ds_list.h, 106
- ds_map, 14
 - ds_map.h, 110
 - hash_size, 14
 - lists, 14
- ds_map.c
 - ds_map_destroy, 108
 - ds_map_get_value, 108
 - ds_map_init, 108
 - ds_map_insert, 108
 - ds_map_print_all, 109
- ds_map.h
 - ds_map, 110
 - ds_map_destroy, 110
 - ds_map_get_value, 110
 - ds_map_init, 110
 - ds_map_insert, 111
 - ds_map_print_all, 111
- ds_map_destroy
 - ds_map.c, 108
 - ds_map.h, 110
- ds_map_get_value
 - ds_map.c, 108
 - ds_map.h, 110
- ds_map_init
 - ds_map.c, 108
 - ds_map.h, 110
- ds_map_insert
 - ds_map.c, 108
 - ds_map.h, 111
- ds_map_print_all
 - ds_map.c, 109
 - ds_map.h, 111
- ds_map_str, 15
 - ds_map_str.h, 114
 - hash_size, 15
 - lists, 15
- ds_map_str.c
 - ds_map_str_destroy, 112
 - ds_map_str_get_value, 112
 - ds_map_str_init, 112
 - ds_map_str_insert, 113
- ds_map_str.h
 - ds_map_str, 114
 - ds_map_str_destroy, 114
 - ds_map_str_get_value, 114
 - ds_map_str_init, 115
 - ds_map_str_insert, 115
- ds_map_str_destroy
 - ds_map_str.c, 112
 - ds_map_str.h, 114
- ds_map_str_get_value
 - ds_map_str.c, 112
 - ds_map_str.h, 114
- ds_map_str_init
 - ds_map_str.c, 112
 - ds_map_str.h, 115
- ds_map_str_insert
 - ds_map_str.c, 113
 - ds_map_str.h, 115
- ds_record, 16
 - ds_record.h, 121
 - fields, 16
- ds_record.c
 - ds_record_clear, 117
 - ds_record_create, 117
 - ds_record_destroy, 117
 - ds_record_destructor, 117
 - ds_record_get_field, 117
 - ds_record_get_next_data, 118
 - ds_record_make_delim_string, 118
 - ds_record_make_values_string, 118
 - ds_record_seek_start, 118
 - ds_record_set_field, 119
 - ds_record_size, 119
 - ds_record_tokenize, 119

- ds_record.h
 - ds_record, [121](#)
 - ds_record_clear, [121](#)
 - ds_record_create, [121](#)
 - ds_record_destroy, [121](#)
 - ds_record_destructor, [122](#)
 - ds_record_get_field, [122](#)
 - ds_record_get_next_data, [122](#)
 - ds_record_make_delim_string, [122](#)
 - ds_record_make_values_string, [122](#)
 - ds_record_seek_start, [123](#)
 - ds_record_set_field, [123](#)
 - ds_record_size, [123](#)
 - ds_record_tokenize, [123](#)
- ds_record_clear
 - ds_record.c, [117](#)
 - ds_record.h, [121](#)
- ds_record_create
 - ds_record.c, [117](#)
 - ds_record.h, [121](#)
- ds_record_destroy
 - ds_record.c, [117](#)
 - ds_record.h, [121](#)
- ds_record_destructor
 - ds_record.c, [117](#)
 - ds_record.h, [122](#)
- ds_record_get_field
 - ds_record.c, [117](#)
 - ds_record.h, [122](#)
- ds_record_get_next_data
 - ds_record.c, [118](#)
 - ds_record.h, [122](#)
- ds_record_make_delim_string
 - ds_record.c, [118](#)
 - ds_record.h, [122](#)
- ds_record_make_values_string
 - ds_record.c, [118](#)
 - ds_record.h, [122](#)
- ds_record_seek_start
 - ds_record.c, [118](#)
 - ds_record.h, [123](#)
- ds_record_set_field
 - ds_record.c, [119](#)
 - ds_record.h, [123](#)
- ds_record_size
 - ds_record.c, [119](#)
 - ds_record.h, [123](#)
- ds_record_tokenize
 - ds_record.c, [119](#)
 - ds_record.h, [123](#)
- ds_recordset, [17](#)
 - ds_recordset.h, [129](#)
 - field_lengths, [17](#)
 - headers, [17](#)
 - num_fields, [17](#)
 - records, [17](#)
 - types, [18](#)
- ds_recordset.c
 - ds_recordset_add_record, [125](#)
 - ds_recordset_create, [125](#)
 - ds_recordset_destroy, [125](#)
 - ds_recordset_get_next_insert_query, [126](#)
 - ds_recordset_get_text_report, [126](#)
 - ds_recordset_next_record, [126](#)
 - ds_recordset_num_fields, [126](#)
 - ds_recordset_num_records, [127](#)
 - ds_recordset_seek_start, [127](#)
 - ds_recordset_set_headers, [127](#)
 - ds_recordset_set_type, [127](#)
- ds_recordset.h
 - ds_recordset, [129](#)
 - ds_recordset_add_record, [129](#)
 - ds_recordset_create, [129](#)
 - ds_recordset_destroy, [130](#)
 - ds_recordset_get_next_insert_query, [130](#)
 - ds_recordset_get_text_report, [130](#)
 - ds_recordset_next_record, [130](#)
 - ds_recordset_num_fields, [130](#)
 - ds_recordset_num_records, [131](#)
 - ds_recordset_seek_start, [131](#)
 - ds_recordset_set_headers, [131](#)
 - ds_recordset_set_type, [131](#)
- ds_recordset_add_record
 - ds_recordset.c, [125](#)
 - ds_recordset.h, [129](#)
- ds_recordset_create
 - ds_recordset.c, [125](#)
 - ds_recordset.h, [129](#)
- ds_recordset_destroy
 - ds_recordset.c, [125](#)
 - ds_recordset.h, [130](#)
- ds_recordset_get_next_insert_query
 - ds_recordset.c, [126](#)
 - ds_recordset.h, [130](#)
- ds_recordset_get_text_report
 - ds_recordset.c, [126](#)
 - ds_recordset.h, [130](#)
- ds_recordset_next_record
 - ds_recordset.c, [126](#)
 - ds_recordset.h, [130](#)
- ds_recordset_num_fields
 - ds_recordset.c, [126](#)
 - ds_recordset.h, [130](#)
- ds_recordset_num_records
 - ds_recordset.c, [127](#)
 - ds_recordset.h, [131](#)
- ds_recordset_seek_start
 - ds_recordset.c, [127](#)
 - ds_recordset.h, [131](#)
- ds_recordset_set_headers
 - ds_recordset.c, [127](#)
 - ds_recordset.h, [131](#)
- ds_recordset_set_type
 - ds_recordset.c, [127](#)
 - ds_recordset.h, [131](#)
- ds_report, [18](#)

- created_time, 19
- ds_report.h, 135
- headers, 19
- report_text, 19
- title, 19
- ds_report.c
 - _XOPEN_SOURCE, 133
 - ds_report_add_header, 133
 - ds_report_create, 133
 - ds_report_destroy, 133
 - ds_report_print_text_report, 133
 - ds_report_set_report_text, 134
 - ds_report_set_title, 134
- ds_report.h
 - ds_report, 135
 - ds_report_add_header, 135
 - ds_report_create, 136
 - ds_report_destroy, 136
 - ds_report_print_text_report, 136
 - ds_report_set_report_text, 136
 - ds_report_set_title, 136
- ds_report_add_header
 - ds_report.c, 133
 - ds_report.h, 135
- ds_report_create
 - ds_report.c, 133
 - ds_report.h, 136
- ds_report_destroy
 - ds_report.c, 133
 - ds_report.h, 136
- ds_report_print_text_report
 - ds_report.c, 133
 - ds_report.h, 136
- ds_report_set_report_text
 - ds_report.c, 134
 - ds_report.h, 136
- ds_report_set_title
 - ds_report.c, 134
 - ds_report.h, 136
- ds_str, 19
 - capacity, 19
 - data, 19
 - ds_str.h, 149
 - length, 19
- ds_str.c
 - ds_str_assign, 139
 - ds_str_assign_cstr, 139
 - ds_str_char_at_index, 139
 - ds_str_clear, 139
 - ds_str_compare, 140
 - ds_str_compare_cstr, 140
 - ds_str_concat, 140
 - ds_str_concat_cstr, 140
 - ds_str_create, 141
 - ds_str_create_direct, 141
 - ds_str_create_sprintf, 141
 - ds_str_cstr, 141
 - ds_str_decorate, 142
 - ds_str_destroy, 142
 - ds_str_destructor, 142
 - ds_str_doubleval, 142
 - ds_str_dup, 143
 - ds_str_getline, 143
 - ds_str_hash, 143
 - ds_str_intval, 143
 - ds_str_is_alnum, 144
 - ds_str_is_empty, 144
 - ds_str_length, 144
 - ds_str_size_to_fit, 144
 - ds_str_split, 145
 - ds_str_strchr, 145
 - ds_str_substr_left, 145
 - ds_str_substr_right, 145
 - ds_str_trim, 146
 - ds_str_trim_leading, 146
 - ds_str_trim_trailing, 146
 - ds_str_trunc, 146
- ds_str.h
 - ds_str, 149
 - ds_str_assign, 149
 - ds_str_assign_cstr, 149
 - ds_str_char_at_index, 149
 - ds_str_clear, 150
 - ds_str_compare, 150
 - ds_str_compare_cstr, 150
 - ds_str_concat, 150
 - ds_str_concat_cstr, 150
 - ds_str_create, 151
 - ds_str_create_direct, 151
 - ds_str_create_sprintf, 151
 - ds_str_cstr, 152
 - ds_str_decorate, 152
 - ds_str_destroy, 152
 - ds_str_destructor, 152
 - ds_str_doubleval, 152
 - ds_str_dup, 153
 - ds_str_getline, 153
 - ds_str_hash, 153
 - ds_str_intval, 153
 - ds_str_is_alnum, 154
 - ds_str_is_empty, 154
 - ds_str_length, 154
 - ds_str_size_to_fit, 154
 - ds_str_split, 155
 - ds_str_strchr, 155
 - ds_str_substr_left, 155
 - ds_str_substr_right, 155
 - ds_str_trim, 156
 - ds_str_trim_leading, 156
 - ds_str_trim_trailing, 156
 - ds_str_trunc, 156
- ds_str_assign
 - ds_str.c, 139
 - ds_str.h, 149
- ds_str_assign_cstr
 - ds_str.c, 139

- [ds_str.h](#), [149](#)
- [ds_str_char_at_index](#)
 - [ds_str.c](#), [139](#)
 - [ds_str.h](#), [149](#)
- [ds_str_clear](#)
 - [ds_str.c](#), [139](#)
 - [ds_str.h](#), [150](#)
- [ds_str_compare](#)
 - [ds_str.c](#), [140](#)
 - [ds_str.h](#), [150](#)
- [ds_str_compare_cstr](#)
 - [ds_str.c](#), [140](#)
 - [ds_str.h](#), [150](#)
- [ds_str_concat](#)
 - [ds_str.c](#), [140](#)
 - [ds_str.h](#), [150](#)
- [ds_str_concat_cstr](#)
 - [ds_str.c](#), [140](#)
 - [ds_str.h](#), [150](#)
- [ds_str_create](#)
 - [ds_str.c](#), [141](#)
 - [ds_str.h](#), [151](#)
- [ds_str_create_direct](#)
 - [ds_str.c](#), [141](#)
 - [ds_str.h](#), [151](#)
- [ds_str_create_sprintf](#)
 - [ds_str.c](#), [141](#)
 - [ds_str.h](#), [151](#)
- [ds_str_cstr](#)
 - [ds_str.c](#), [141](#)
 - [ds_str.h](#), [152](#)
- [ds_str_decorate](#)
 - [ds_str.c](#), [142](#)
 - [ds_str.h](#), [152](#)
- [ds_str_destroy](#)
 - [ds_str.c](#), [142](#)
 - [ds_str.h](#), [152](#)
- [ds_str_destructor](#)
 - [ds_str.c](#), [142](#)
 - [ds_str.h](#), [152](#)
- [ds_str_doubleval](#)
 - [ds_str.c](#), [142](#)
 - [ds_str.h](#), [152](#)
- [ds_str_dup](#)
 - [ds_str.c](#), [143](#)
 - [ds_str.h](#), [153](#)
- [ds_str_getline](#)
 - [ds_str.c](#), [143](#)
 - [ds_str.h](#), [153](#)
- [ds_str_hash](#)
 - [ds_str.c](#), [143](#)
 - [ds_str.h](#), [153](#)
- [ds_str_intval](#)
 - [ds_str.c](#), [143](#)
 - [ds_str.h](#), [153](#)
- [ds_str_is_alnum](#)
 - [ds_str.c](#), [144](#)
 - [ds_str.h](#), [154](#)
- [ds_str_is_empty](#)
 - [ds_str.c](#), [144](#)
 - [ds_str.h](#), [154](#)
- [ds_str_length](#)
 - [ds_str.c](#), [144](#)
 - [ds_str.h](#), [154](#)
- [ds_str_size_to_fit](#)
 - [ds_str.c](#), [144](#)
 - [ds_str.h](#), [154](#)
- [ds_str_split](#)
 - [ds_str.c](#), [145](#)
 - [ds_str.h](#), [155](#)
- [ds_str_strchr](#)
 - [ds_str.c](#), [145](#)
 - [ds_str.h](#), [155](#)
- [ds_str_substr_left](#)
 - [ds_str.c](#), [145](#)
 - [ds_str.h](#), [155](#)
- [ds_str_substr_right](#)
 - [ds_str.c](#), [145](#)
 - [ds_str.h](#), [155](#)
- [ds_str_trim](#)
 - [ds_str.c](#), [146](#)
 - [ds_str.h](#), [156](#)
- [ds_str_trim_leading](#)
 - [ds_str.c](#), [146](#)
 - [ds_str.h](#), [156](#)
- [ds_str_trim_trailing](#)
 - [ds_str.c](#), [146](#)
 - [ds_str.h](#), [156](#)
- [ds_str_trunc](#)
 - [ds_str.c](#), [146](#)
 - [ds_str.h](#), [156](#)
- [ds_vector](#), [20](#)
 - [current](#), [20](#)
 - [data](#), [20](#)
 - [data_destructor](#), [20](#)
 - [ds_vector.h](#), [161](#)
 - [free_on_delete](#), [20](#)
 - [size](#), [20](#)
- [ds_vector.c](#)
 - [ds_vector_clear](#), [158](#)
 - [ds_vector_create](#), [158](#)
 - [ds_vector_destroy](#), [158](#)
 - [ds_vector_destructor](#), [158](#)
 - [ds_vector_element](#), [158](#)
 - [ds_vector_get_next_data](#), [159](#)
 - [ds_vector_seek_start](#), [159](#)
 - [ds_vector_set](#), [159](#)
 - [ds_vector_size](#), [159](#)
- [ds_vector.h](#)
 - [ds_vector](#), [161](#)
 - [ds_vector_clear](#), [161](#)
 - [ds_vector_create](#), [161](#)
 - [ds_vector_destroy](#), [162](#)
 - [ds_vector_destructor](#), [162](#)
 - [ds_vector_element](#), [162](#)
 - [ds_vector_get_next_data](#), [162](#)

- ds_vector_seek_start, 163
 - ds_vector_set, 163
 - ds_vector_size, 163
- ds_vector_clear
 - ds_vector.c, 158
 - ds_vector.h, 161
- ds_vector_create
 - ds_vector.c, 158
 - ds_vector.h, 161
- ds_vector_destroy
 - ds_vector.c, 158
 - ds_vector.h, 162
- ds_vector_destructor
 - ds_vector.c, 158
 - ds_vector.h, 162
- ds_vector_element
 - ds_vector.c, 158
 - ds_vector.h, 162
- ds_vector_get_next_data
 - ds_vector.c, 159
 - ds_vector.h, 162
- ds_vector_seek_start
 - ds_vector.c, 159
 - ds_vector.h, 163
- ds_vector_set
 - ds_vector.c, 159
 - ds_vector.h, 163
- ds_vector_size
 - ds_vector.c, 159
 - ds_vector.h, 163
- field_lengths
 - ds_recordset, 17
- fields
 - ds_record, 16
- free_on_delete
 - ds_list, 12
 - ds_vector, 20
- get_cmdline_options
 - gl_db_config.c, 186
 - gl_db_config.h, 187
 - gl_reports_config.c, 191
 - gl_reports_config.h, 192
- get_configuration
 - gl_config.c, 174
 - gl_config.h, 176
- gl_config.c
 - get_configuration, 174
 - params_free, 174
 - params_init, 174
- gl_config.h
 - get_configuration, 176
 - params_free, 176
 - params_init, 176
- gl_db_config.c
 - _XOPEN_SOURCE, 185
 - get_cmdline_options, 186
- gl_db_config.h
 - get_cmdline_options, 187
- gl_db_main.c
 - main, 189
 - print_help_message, 189
 - print_usage_message, 189
 - print_version_message, 189
- gl_error_quit
 - gl_errors.c, 177
 - gl_errors.h, 178
- gl_errors.c
 - gl_error_quit, 177
- gl_errors.h
 - gl_error_quit, 178
- gl_log_msg
 - gl_logging.c, 180
 - gl_logging.h, 182
- gl_logging.c
 - gl_log_msg, 180
 - gl_set_logging, 180
- gl_logging.h
 - gl_log_msg, 182
 - gl_set_logging, 182
- gl_login.c
 - login, 183
- gl_login.h
 - login, 184
- gl_reports_config.c
 - _XOPEN_SOURCE, 190
 - get_cmdline_options, 191
- gl_reports_config.h
 - get_cmdline_options, 192
- gl_set_logging
 - gl_logging.c, 180
 - gl_logging.h, 182
- hash_size
 - ds_map, 14
 - ds_map_str, 15
- head
 - ds_list, 13
- headers
 - ds_recordset, 17
 - ds_report, 19
- hostname
 - params, 22
- key
 - ds_kvpair, 11
 - kv_pair_node, 21
- kv_pair_node, 21
 - key, 21
 - next, 21
 - value, 21
- length
 - ds_list, 13
 - ds_str, 19
- lib/database/database.h, 23
- lib/database/db_connection.h, 24

- lib/database/db_currenttb.c, 25
- lib/database/db_currenttb.h, 27
- lib/database/db_entities.c, 29
- lib/database/db_entities.h, 30
- lib/database/db_internal.h, 33
- lib/database/db_jelines.c, 33
- lib/database/db_jelines.h, 35
- lib/database/db_jes.c, 36
- lib/database/db_jes.h, 38
- lib/database/db_jesrcs.c, 40
- lib/database/db_jesrcs.h, 42
- lib/database/db_nomaccts.c, 43
- lib/database/db_nomaccts.h, 44
- lib/database/db_query.h, 46
- lib/database/db_reporting.c, 47
- lib/database/db_reporting.h, 48
- lib/database/db_sampledata.c, 49
- lib/database/db_sampledata.h, 50
- lib/database/db_sql.h, 51
- lib/database/db_standingdata.c, 58
- lib/database/db_standingdata.h, 59
- lib/database/db_structure.c, 60
- lib/database/db_structure.h, 62
- lib/database/db_users.c, 63
- lib/database/db_users.h, 64
- lib/database/dummy/db_dummy_create_entities_table_sql.c, 65
- lib/database/dummy/db_dummy_create_users_table_sql.c, 66
- lib/database/dummy/db_dummy_drop_entities_table_sql.c, 67
- lib/database/dummy/db_dummy_drop_users_table_sql.c, 67
- lib/database/dummy/db_dummy_general.c, 68
- lib/database/dummy/db_dummy_list_entities_report_sql.c, 70
- lib/database/dummy/db_dummy_list_users_report_sql.c, 71
- lib/database/mysql/db_mysql_all_jes_number_report_sql.c, 71
- lib/database/mysql/db_mysql_all_jes_report_sql.c, 72
- lib/database/mysql/db_mysql_check_total_entity_report_sql.c, 72
- lib/database/mysql/db_mysql_check_total_report_sql.c, 73
- lib/database/mysql/db_mysql_create_all_jes_view_sql.c, 74
- lib/database/mysql/db_mysql_create_check_total_view_sql.c, 74
- lib/database/mysql/db_mysql_create_current_trial_balance_view_sql.c, 75
- lib/database/mysql/db_mysql_create_entities_table_sql.c, 75
- lib/database/mysql/db_mysql_create_jelines_table_sql.c, 76
- lib/database/mysql/db_mysql_create_jes_table_sql.c, 77
- lib/database/mysql/db_mysql_create_jesrcs_table_sql.c, 77
- lib/database/mysql/db_mysql_create_nomaccts_table_sql.c, 78
- lib/database/mysql/db_mysql_create_standingdata_table_sql.c, 78
- lib/database/mysql/db_mysql_create_users_table_sql.c, 79
- lib/database/mysql/db_mysql_current_trial_balance_entity_report_sql.c, 80
- lib/database/mysql/db_mysql_current_trial_balance_report_sql.c, 80
- lib/database/mysql/db_mysql_drop_all_jes_view_sql.c, 81
- lib/database/mysql/db_mysql_drop_check_total_view_sql.c, 81
- lib/database/mysql/db_mysql_drop_current_trial_balance_view_sql.c, 82
- lib/database/mysql/db_mysql_drop_entities_table_sql.c, 83
- lib/database/mysql/db_mysql_drop_jelines_table_sql.c, 83
- lib/database/mysql/db_mysql_drop_jes_table_sql.c, 84
- lib/database/mysql/db_mysql_drop_jesrcs_table_sql.c, 84
- lib/database/mysql/db_mysql_drop_nomaccts_table_sql.c, 85
- lib/database/mysql/db_mysql_drop_standingdata_table_sql.c, 86
- lib/database/mysql/db_mysql_drop_users_table_sql.c, 86
- lib/database/mysql/db_mysql_entity_name_from_id_sql.c, 87
- lib/database/mysql/db_mysql_general.c, 87
- lib/database/mysql/db_mysql_list_entities_report_sql.c, 89
- lib/database/mysql/db_mysql_list_jelines_report_sql.c, 90
- lib/database/mysql/db_mysql_list_jes_report_sql.c, 90
- lib/database/mysql/db_mysql_list_jesrcs_report_sql.c, 91
- lib/database/mysql/db_mysql_list_nomaccts_report_sql.c, 92
- lib/database/mysql/db_mysql_list_users_report_sql.c, 92
- lib/database/mysql/db_mysql_show_standingdata_report_sql.c, 93
- lib/datastruct/data_structures.h, 94
- lib/datastruct/ds_fieldtypes.h, 94
- lib/datastruct/ds_kvpair.h, 95
- lib/datastruct/ds_list.c, 98
- lib/datastruct/ds_list.h, 102
- lib/datastruct/ds_map.c, 106
- lib/datastruct/ds_map.h, 109
- lib/datastruct/ds_map_str.c, 111
- lib/datastruct/ds_map_str.h, 113
- lib/datastruct/ds_record.c, 115
- lib/datastruct/ds_record.h, 119

- lib/datastruct/ds_recordset.c, [124](#)
- lib/datastruct/ds_recordset.h, [127](#)
- lib/datastruct/ds_report.c, [132](#)
- lib/datastruct/ds_report.h, [134](#)
- lib/datastruct/ds_str.c, [137](#)
- lib/datastruct/ds_str.h, [146](#)
- lib/datastruct/ds_vector.c, [156](#)
- lib/datastruct/ds_vector.h, [160](#)
- lib/file_ops/config_file_read.c, [163](#)
- lib/file_ops/config_file_read.h, [166](#)
- lib/file_ops/delim_file_read.c, [169](#)
- lib/file_ops/delim_file_read.h, [171](#)
- lib/file_ops/file_ops.h, [172](#)
- lib/gl_general/gl_config.c, [173](#)
- lib/gl_general/gl_config.h, [175](#)
- lib/gl_general/gl_errors.c, [177](#)
- lib/gl_general/gl_errors.h, [178](#)
- lib/gl_general/gl_general.h, [178](#)
- lib/gl_general/gl_logging.c, [179](#)
- lib/gl_general/gl_logging.h, [181](#)
- lib/gl_general/gl_login.c, [182](#)
- lib/gl_general/gl_login.h, [183](#)
- lists
 - ds_map, [14](#)
 - ds_map_str, [15](#)
- login
 - gl_login.c, [183](#)
 - gl_login.h, [184](#)
- MAX_BUFFER_SIZE
 - config_file_read.c, [165](#)
- MAX_LINE_SIZE
 - delim_file_read.c, [170](#)
- main
 - gl_db_main.c, [189](#)
- main_mss
 - db_mysql_general.c, [89](#)
- next
 - ds_list_element, [13](#)
 - kv_pair_node, [21](#)
- num_fields
 - ds_recordset, [17](#)
- params, [21](#)
 - database, [22](#)
 - hostname, [22](#)
 - password, [22](#)
 - username, [22](#)
- params_free
 - gl_config.c, [174](#)
 - gl_config.h, [176](#)
- params_init
 - gl_config.c, [174](#)
 - gl_config.h, [176](#)
- password
 - params, [22](#)
- previous
 - ds_list_element, [14](#)
- print_help_message
 - gl_db_main.c, [189](#)
- print_usage_message
 - gl_db_main.c, [189](#)
- print_version_message
 - gl_db_main.c, [189](#)
- progs/gl_db/gl_db_config.c, [184](#)
- progs/gl_db/gl_db_config.h, [186](#)
- progs/gl_db/gl_db_main.c, [188](#)
- progs/gl_reports/gl_reports_config.c, [189](#)
- progs/gl_reports/gl_reports_config.h, [191](#)
- records
 - ds_recordset, [17](#)
- report_text
 - ds_report, [19](#)
- size
 - ds_vector, [20](#)
- tail
 - ds_list, [13](#)
- title
 - ds_report, [19](#)
- types
 - ds_recordset, [18](#)
- username
 - params, [22](#)
- value
 - ds_kvpair, [11](#)
 - kv_pair_node, [21](#)